

HISTOLOGISEN KUDOSNÄYTTEEN PINTA-ALAN MITTAAMINEN DIGITAALISEN KUVA-ANALYYSIN MENETELMIN

LK OSKARI JÄÄSKELÄINEN
SYVENTÄVIEN OPINTOJEN KIRJALLINEN TYÖ
TAMPEREEN YLIOPISTO
LÄÄKETIETEEN YKSIKKÖ
TOUKOKUU 2013

Tampereen yliopisto
Lääketieteen yksikkö

JÄÄSKELÄINEN OSKARI: HISTOLOGISEN KUDOSNÄYTTEEN PINTA-ALAN MITTAAMINEN
DIGITAALISEN KUVA-ANALYYSIN MENETELMIN

Kirjallinen työ, 45s.
Ohjaaja: dosentti Ivana Kholová

Toukokuu 2013

Avainsanat: digitaalinen patologia, histologia, morfometria, virtuaalimikroskopia

Tämän tutkimuksen tarkoituksena oli kehittää digitaalisen kuva-analyysin menetelmä histologisen kudoksen näytteen pinta-alan mittaamiseksi. Menetelmä tulee käyttöön muihin meneillä oleviin tutkimuksiin, joissa selvitetään eteisvärinän patofysiologiaa tutkimalla sydämeen tulevien laskimoiden ja erityisesti niiden hermotuksen morfologiaa.

Näytelasien digitointiin käytettiin Aperio ScanScope® XT –skanneria ja virtuaalinäytelasien katseluun JVSView –ohjelmaa. Itse pinta-alan mittaava algoritmi kehitettiin ja toteutettiin avoimen lähdekoodin ImageJ –ohjelman makrona.

Tuloksena saatiin algoritmi, joka on tarkka ja huomattavasti vastaavalla tarkkuudella tehtyä manuaalista mittausta nopeampi. Lisäksi tehtiin yksityiskohtaiset työohjeet virtuaalinäytelasien mittaamista varten.

SISÄLLYS

1 Johdanto.....	1
2 Aineisto	2
3 Menetelmän kuvaus.....	3
4 Apumakrot	6
5 Algoritmin kuvaus.....	7
6 Työohjeet	9
6.1 Tarvittavat ohjelmat.....	9
6.2 Ohjelmien asennus.....	10
6.2.1 JVSview	10
6.2.2 ImageJ	12
6.2.3 Ohjelmien konfigurointi	19
6.3 Virtuaalilasin katselu	22
6.4 Kuvien siirto ImageJ-ohjelmaan	23
6.5 Pinta-alan mittaaminen.....	27
7 Pohdinta	33
Viitteet	36
Liite 1.....	37
Taulukot	40
Kuvat	41

1 JOHDANTO

Histologia eli kudosophi on nykymuotoisen patologian perusta. Aiemmin kudoksia ja niiden muutoksia oli tutkittu silmämääräisesti kirurgian ja ruumiinavausten yhteydessä, mutta mikroskoopin keksiminen johti solujen löytämiseen. Saksalainen lääkäri Rudolf Virchow (1821–1902) kehitti soluihin perustuvan tautiopin, joka on modernin patologian perusta. (Laitio 1992)

Diagnostinen patologia perustuu suurimmaksi osaksi solujen morfologiaan valomikroskopiassa. Lisäksi immunohistokemiallisten menetelmien merkitys on kasvanut. Morfometria on histologian osa-alue, joka käsittää kudosten, solujen ja soluorganellien mittaamisen. Morfometriaa sovelletaan lähinnä tutkimuskäytössä sekä joissain diagnostisissa ongelmissa (esim. Koskinen 2005).

Valomikroskopian käytetyin muoto on kirkaskenttämikroskopia. Muita muotoja ovat fluoresenssimikroskopia, vaihekontrastimikroskopia ja konfokaalimikroskopia. Virtuaalimikroskopialla tarkoitetaan histologisten, hematologisten ja sytologisten näytteiden skannausta digitaaliseen muotoon ja näin saadun kuvainformaation käsittelyä ja analysointia tietokoneavusteisesti. Analyysi voidaan tehdä visuaalisesti tietokoneen näytöltä tai ohjelmallisesti. Myös termiä digitaalinen patologia on käytetty synonyymina virtuaalimikroskopialle, mutta digitaalinen patologia käsittää laajemman alueen sisältäen myös esimerkiksi laboratoriotietojärjestelmät. (Weinstein 2005)

2 AINEISTO

Aineistona käytettiin osaa aikaisemmin kerätystä obduktioaineistosta. Aineistossa on näytteitä eteisvärinäpotilaiden ja terveiden verrokkien sydämien vasempaan ja oikeaan eteiseen verta tuovista suurista laskimoista ja eteisten ja laskimoiden liitoskohdista. Kholová ja Kautzner ovat aiemmin tutkineet aineistosta eteisten myokardiumin ja sen laskimoiden seinämiin jatkuvien hihojen morfologiaa ja morfometriaa sekä yhteyttä eteisvärinään. (Kholová 2003 & 2004)

Sydämet kerättiin niin, että mukaan tulivat keuhkolaskimot aina keuhkon hilukseen asti. Pulmonaarilaskimot irrotettiin vasemmasta eteisestä ostiumin tasolta, joka määriteltiin jyrkäksi muutokseksi pulmonaarilaskimon ja vasemman eteisen välillä. Laskimon avattiin pitkittäin ja levitettiin tasoksi. (Kholová 2003)

Ylä- ja alaonttolaskimot kerättiin vastaavalla tavalla määrittämällä makroskooppisesti onttolaskimon ja eteisen junktio ja irrottamalla suoni sen kohdalta. Myös ylä- ja alaonttolaskimot avattiin pitkittäin ja levitettiin tasoksi. (Kholová 2004)

Kaikki suonet fiksoitiin formaliinissa standardiprotokollan mukaan. Avattujen suonten päistä leikattiin poikittaiset näytepalat, jotka vastaavat ehjän suonen sirkulaarista leikettä. Loppuosasta suonta leikattiin longitudinaaliset suonen akselin suuntaiset näytepalat. Näytepalat valettiin parafiiniblokkeihin. (Kholová 2003 & 2004)

3 MENETELMÄN KUVAUS

Tämän projektin tarkoituksena oli kehittää nopea ja tarkka digitaalisen kuva-analyysin menetelmä histologisen kudoksen pinta-alan mittaamiseksi. Menetelmä tulee käyttöön muihin meneillä oleviin tutkimuksiin, joissa selvitetään eteisvärinän patofysiologiaa tutkimalla sydämeen tulevien laskimoiden ja erityisesti niiden hermotuksen morfologiaa. Eteisvärinän syntymekanismia ei vielä tunneta, vaikka sitä on tutkittu jo yli sata vuotta (Jalife 2011). Tästä syystä myöskään tehokasta hoitoa ei ole (Eteisvärinä: Käypä hoito – suositus 2011).

Parafiiniblokeissa olevista näytteistä leikattiin näytelaseille uudet 5µm paksuiset leikkeet. Hermojen visualisointia varten morfometriin tutkimuksiin oli valittu kolme immunohistokemiallista värjäystä. Vasta-aineina näissä värjäyksissä olivat Anti-Choline Acetyltransferase (ChAT), Anti-Tyrosine Hydroxylase (ATH) ja Anti-Growth Associated Protein-43 (GAP-43). Näistä ATH kiinnittyy adrenergisiin hermoihin ja ChAT kolinergisiin. GAP-43 on aksonaalinen kasvutekijä. Entsyyminä oli piparjuuriperoksidaasi (horseradishperoksidas, HRP), kromogeeni oli 3,3'-diaminobenzidiini (DAB) ja taustaväri hematoksyliini. Menetelmä on yleisesti käytetty (Dabbs 2006). Värjäykset tehtiin Ventana BenchMark® -värjäysautomaatilla Tampereen Yliopiston patologian laboratoriossa.

TAULUKKO 1: IMMUNOHISTOKEMIAALLISET VASTA-AINEET

Vasta-aine	Valmistaja	Laimennus
Anti-Choline Acetyltransferase	Millipore	1:300
Anti-Tyrosine Hydroxylase	Millipore	1:100
Anti-Growth Associated Protein-43	Millipore	1:100

Varsinainen virtuaalimikroskopian osuus alkoi näytelasien kuvaamisella digitaaliseen muotoon. Tämä on olennainen ja tarkkuutta vaativa työvaihe, sillä huonolla kuvaamisella kadotettua informaatiota ei enää millään kuvamanipulaatiolla saada esiin. Koko näytelasin digitointi tuottaa valtavan määrän informaatiota. Tavallisin näytelasin koko on 25 x 75 mm, ja tavallinen skannausresoluutio on 0,25 – 0,5 μm pikseliä kohden. Tästä voi helposti laskea kokonaisen virtuaalinäytelasin tietomäärän olevan useita kymmeniä tai jopa satoja gigatavuja. Käytännön syistä virtuaalinäytelasit on pakattava häviöllisellä algoritmilla. JPEG2000 on tähän hyvin soveltuva standardi (Tuominen 2008).

Näytelasit skannattiin Aperio ScanScope XT[®] -skannerilla 20X-objektiivilla. Skannerilta saadaan virtuaalinäytelasit JPEG2000-muodossa. Näytelasien ja skannauksen laatu varmistettiin käymällä jokainen virtuaalinäytelasi visuaalisesti läpi. Virtuaalinäytelasien katselemiseen käytettiin JVSview- ohjelmaa (Tuominen 2008). Ongelmallisissa tapauksissa näytteet tarvittaessa leikattiin, värjättiin ja skannattiin uudestaan.

Varsinaista analyysia varten kuvat tuli siirtää ImageJ -ohjelmaan. Tätä varten JVSview:ssä on oma liityntärajapinta. Siirto tehtiin resoluutiolla, jossa ImageJ -ohjelmaan saatavan kuvan yksi pikseli vastaa yhtä mikrometriä näytelasilla. Tämä on puolet käytetystä skannausresoluutiosta. Koska siirto ohjelmien välillä tapahtuu näkökenttä kerrallaan, kehitettiin ImageJ -ohjelmaan makro, joka yhdistää oikeassa järjestyksessä kuvatut näkökentät yhdeksi kuvaksi. Tämä oli tärkeää myös pinta-alan mittauksen kannalta, koska kehitetty algoritmi käsittelee eri tavalla näytteen reuna-alueita kuin keskustaa.

Seuraavaksi ImageJ -ohjelmalle määriteltiin näytteen mittakaava, joka siis oli 1000 pikseliä/mm. Myös tätä toimintoa nopeuttamaan kehitettiin makro, joka suoritetaan ennen pinta-alan mittaavaa makroa. Tämän mittakaavatiedon avulla ImageJ pystyy antamaan kuvasta suoritettujen mittausten tulokset todellisissa mittayksiköissä.

Varsinaista pinta-alan mittaamista varten kehitettiin monivaiheinen makro. Ihminen pystyy suhteellisen helposti erottamaan mikä osa kuvasta kuuluu näytteeseen ja mikä ei. Tietokoneohjelmalle on määriteltävä mitkä pikselit kuuluvat näytteeseen ja mitkä eivät kuulu.

Algoritmin ensimmäisessä vaiheessa 24-bittinen värikuva muunnettiin 8-bittiseksi harmaasävykuvaksi valoisuusarvon perusteella. Toisessa vaiheessa kuva kynnystettiin, eli se jaettiin kahteen osaan asettamalla raja-arvo, jonka toisella puolella olevista sävyistä tehtiin puhtaasti mustaa ja toisella puolella olevista valkoista. Tässä vaiheessa kuva on jaettu kahteen osaan. Mustat pikselit edustavat näytettä ja valkoiset taustaa. Seuraavaksi taustan alueelle tulleet merkittävät artefaktat poistettiin manuaalisesti, mikäli niitä oli.

Tässä vaiheessa suurin osa mustista pikseleistä oli näytteen alueella. Ne eivät kuitenkaan kata koko näytettä, koska luonnollisesti näytteen vaaleammat kohdat muuntuivat kynnystettäessä valkoisiksi. Erityisesti rasvasolujen sisältö jää formaliinifikaatiossa tyhjäksi ja siten kuvassa valkoiseksi. Lisäksi taustan alueella on pieniä mustien pikseleiden ryhmiä. Näiden poistamiseksi ja näytteen "reikien" täyttämiseksi algoritmissa käytettiin toistuvia "dilate" eli laajenna -funktioita joita seurasi vastaava määrä erode eli syövytä -funktioita. Lisäksi käytettiin ImageJ:n "fill holes" eli täytä reiät -funktioita. Näillä kuvankäsittelyoperaatioilla saatiin kuvan mustat pikselit hyvin vastaamaan näytteen aluetta. Näytteen pinta-ala saatiin nyt ImageJ:n measure-toiminnolla.

4 APUMAKROT

Kehitystyön tuloksena syntyi kuvatiedostosta kudoksen näytteen pinta-alan mittaava algoritmi, joka on toteutettu ImageJ-ohjelman makrona OJ_Area.ijm. Lisäksi kuvien siirtoa varten kehitettiin kaksi apumakroa, OJ_SetScale.ijm sekä OJ_Montage.ijm. Makrojen lähdekoodi löytyy liitteestä 1.

OJ_SetScale.ijm on ImageJ:n omaa Set Scale -funktion käyttöä nopeuttava apuväline. Se avaa käyttäjälle yksinkertaisen ikkunan, jossa valittavana on JVSView -ohjelmassa käytetty suurennuskerroin. Sen jälkeen se asettaa käyttäjän valinnan perusteella Set Scale -funktiolla kuvalle mittakaavan. Esimerkiksi suurennoksen ”20X” ollessa käytössä, vastaa 1000 pikseliä kuvassa yhtä millimetriä todellisuudessa. ”40X” -suurennoksella käytössä on skannerin natiiviresoluutio (<http://jvsmicroscope.uta.fi/?q=jvsview>), joka Aperio Sanscope XT -skannerin tapauksessa on 0,5 µm / pikseli, eli 2000 pikseliä vastaa yhtä millimetriä.

Virtuaalinäytelasien katseluohjelma JVSView tarjoaa vain rajallisen mahdollisuuden siirtää kuvia analysoitavaksi ImageJ -ohjelmaan. Käytännössä siirto täytyy tehdä näkökenttä kerrallaan, jolloin koko näytteen siirtäminen tuottaa useita kymmeniä erillisiä kuvia ImageJ-ohjelmaan. Koska pinta-alan mittaaminen vaatii yhtenäisen kuvan, täytyy yksittäiset kuvat yhdistää oikeille paikoilleen. OJ_Montage.ijm helpottaa tätä palapeliä. Käyttäjältä kysytään kuvan nimeä (oletus on JVSView ohjelmalta välittyvästä tiedostonimesta muodostettu) sekä sarakkeiden ja rivien määrää. Oletuksena on, että kaikki kuvat ovat pikselitarkasti samankokoisia ja myös pikselitarkasti vierekkäisiä. Kuvat kootaan yhteen käyttämällä ImageJ:n Make montage -toimintoa. Jotta tämä toimisi, on kuvat siirrettävä ImageJ-ohjelmaan oikeassa järjestyksessä: rivi kerrallaan alkaen vasemmasta yläkulmasta.

5 ALGORITMIN KUVAUS

Varsinainen pinta-alan mittausta tapahtuu algoritmilla, joka sisältyy `OJ_Area.ijm` -makroon. Algoritmi on monivaiheinen. Tavoitteena on jakaa kuva kahteen osaan: näyte ja tausta. Toisin sanoen lopputuloksena pitäisi olla yksibittinen eli kaksivärinen kuva. Tällaisen kuvan jokaisen pikselin väri voidaan ilmoittaa yhdellä bitillä. Se voi saada joko arvon 1 tai 0, musta tai valkoinen.

Lähtötilanteena on 24-bittinen RGB-kuva, eli kuva, jossa on kolme värikanavaa (punainen, vihreä ja sininen) ja jokaisen arvo on ilmoitettu 8 bitin avulla. Algoritmin ensimmäisessä vaiheessa tämä muunnetaan 8-bittiseksi harmaasävykuvaksi ImageJ:n sisäänrakennetulla komenolla "8-bit". Tämä muunnos käyttää suoraviivaista kaavaa $gray = (red+green+blue)/3$ (Ferreira 2011).

Seuraavana vaiheena on kuvan jakaminen kahteen osaan raja-arvon perusteella. Raja-arvoistaminen eli thresholding on yleinen tapa segmentoida kuva kahteen tai useampaan osaan. Raja-arvon valitsemiseen on useita menetelmiä, mutta mikään yksittäinen menetelmä ei sovellu kaikkiin tilanteisiin, eikä mikään menetelmä ole täydellinen. (Young 2007) Tähän algoritmiin valittiin visuaalinen manuaalinen menetelmä, eli käyttäjä valitsee raja-arvon, joka visuaalisesti parhaiten erottelee näytteen ja taustan.

Kuva on nyt eroteltu kahteen osaan pikseleiden vaaleuden/tummuuden perusteella. Tämä erottelu ei kuitenkaan vastaa täysin näytteen pinta-alaa, koska myös näytteen alueella on vaaleita kohtia. Osa alueista on värjäytynyt hailakammin kuin muu näyte, ja raja-arvoistaminen muuttaa nämä valkoisiksi. Vaaleiksi jäävät myös mm. rasvasolujen sisältö (rasva poistuu tavanomaisessa kudoksenäytteen prosessoinnissa) ja osa verisuonten luumeneista. Lisäksi värjäyksestä riippuen osa soluista tai solun osista jää värjäytymättä. Seuraavien vaiheiden tarkoitus on täyttää nämä näytteen sisäpuolelle jäävät valkoiset alueet kuitenkin suurentamatta näytteen ulkopuolella olevia mustia alueita. Tähän käytetään

ImageJ:n sisäänrakennettuja "Dilate", "Erode" ja "Fill Holes" -funktioita. "Dilate" laajentaa mustia alueita lisäämällä mustia pikseleitä niiden reunoille, "Erode" puolestaan supistaa niitä vastaavasti (Ferreira 2011, Young 2007). Näiden yhdistäminen oikeassa järjestyksessä täyttää pienet reiät näytteessä ja pehmentää hieman ääri viivoja (Ferreira 2011, Young 2007). Koska molemmat operaatiot käsittelevät vain yhden pikselin levyistä aluetta näytteen reunalla, täytyy niitä toistaa useamman kerran peräkkäin, jotta saadaan toivottu vaikutus. Isommat reiät täytetään "Fill Holes" -funktiolla. Se täyttää valkoiset alueet, jotka ovat joka puolelta rajautuneet mustaan (Ferreira 2011). OJ_Area.ijm -makroon valittu yhdistelmä täyttää hyvin näytteen sisäpuolelle jääneet reiät, mutta ei suurennakaan näytteen ulkomittoja tai taustassa näkyviä pieniä roskia. Lopuksi suoritetaan ImageJ:n "Measure" -funktio, joka laskee mustien pikselien määrän ja mittakaavatiedon avulla muuntaa sen pinta-alaksi.

6 TYÖOHJEET

6.1 TARVITTAVAT OHJELMAT

Menetelmä on kehitetty alla luetelluilla ohjelmilla ja niiden ilmoitetuilla versioilla. On mahdollista saada menetelmä toimimaan muillakin ohjelmilla ja versioilla, mutta mittausten tarkkuuden ja luotettavuuden arviointi tulee tällöin tehdä uudelleen.

Aperio ScanScope® XT skanneri (AperioTechnologies, USA)

Aperio ScanScope Console v. 9.1.0.1567

Aperio ScanScope Controller v. 9.1.0.1567

DirObserver v. 1.0.1-alpha, saatavilla osoitteesta <http://jvsmicroscope.uta.fi/?q=software>

(Aperio ScanScope® XT -skannerin ja sen ohjelmien ja apuohjelmien asennus on tämän ohjeen ulkopuolella.)

JVSview versio 1.2, saatavilla osoitteesta <http://jvsmicroscope.uta.fi/?q=software>

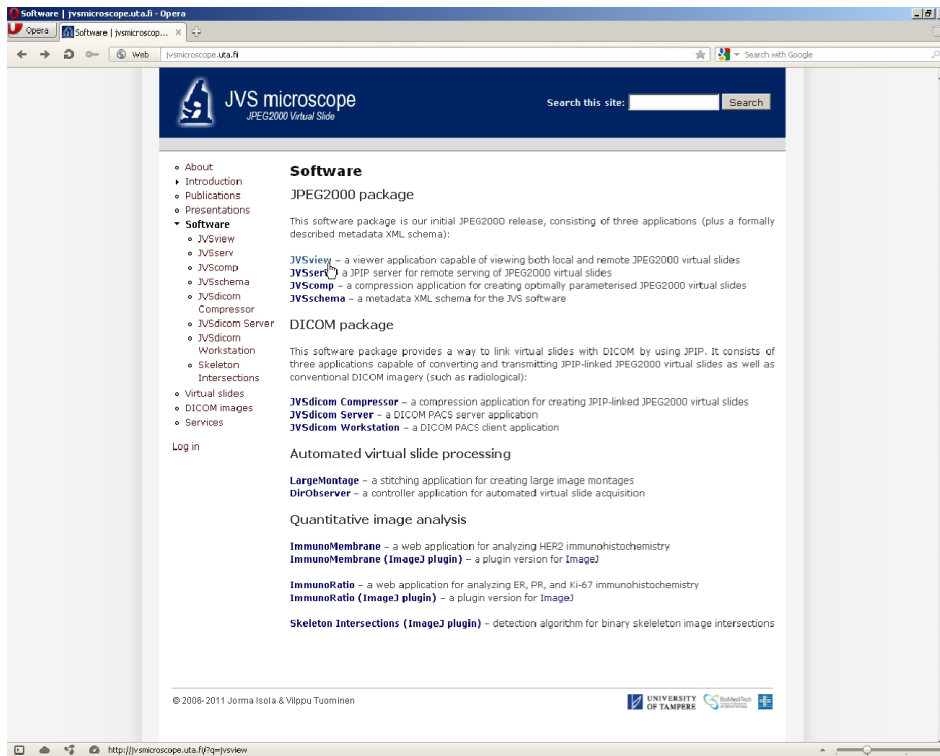
ImageJ, versio 1.45I (20 July 2011), saatavilla osoitteesta <http://imagej.nih.gov/ij/>

ImageJ\plugins\OJ -kansio, jossa tiedostot OJ_Area.ijm, OJ_SetScale.ijm, OJ_Montage.ijm

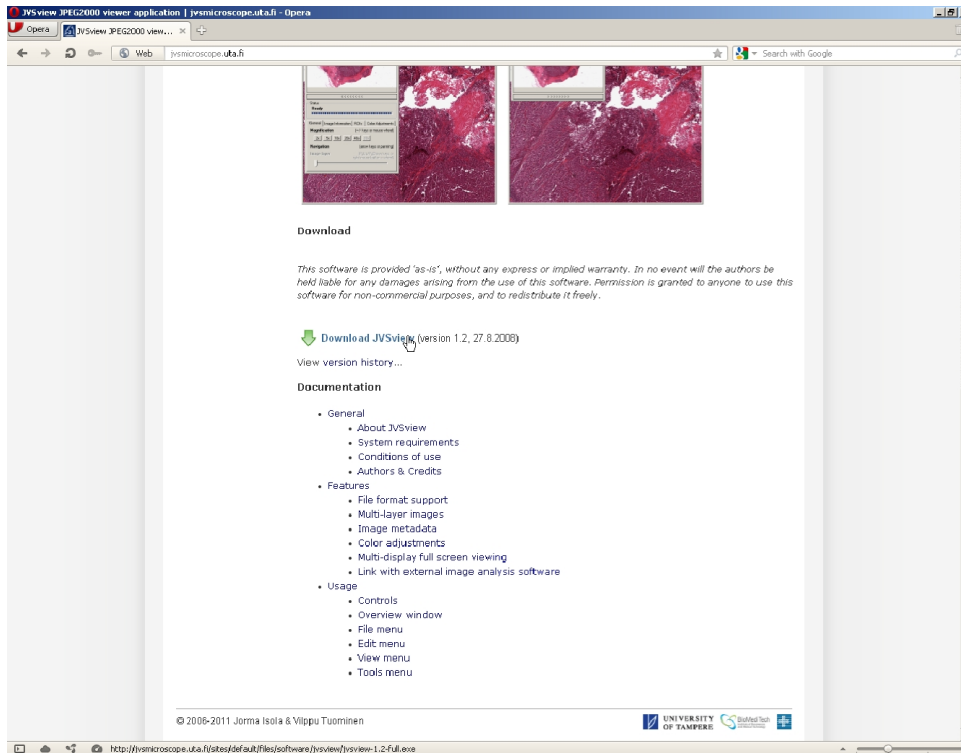
6.2 OHJELMIEN ASENNUS

6.2.1 JVSVIEW

Lataa JVSview osoitteesta <http://jvsmicroscope.uta.fi/?q=software>. (Kuva 1, Kuva 2)



KUVA 1: JVSVIEW:N LATAAMINEN



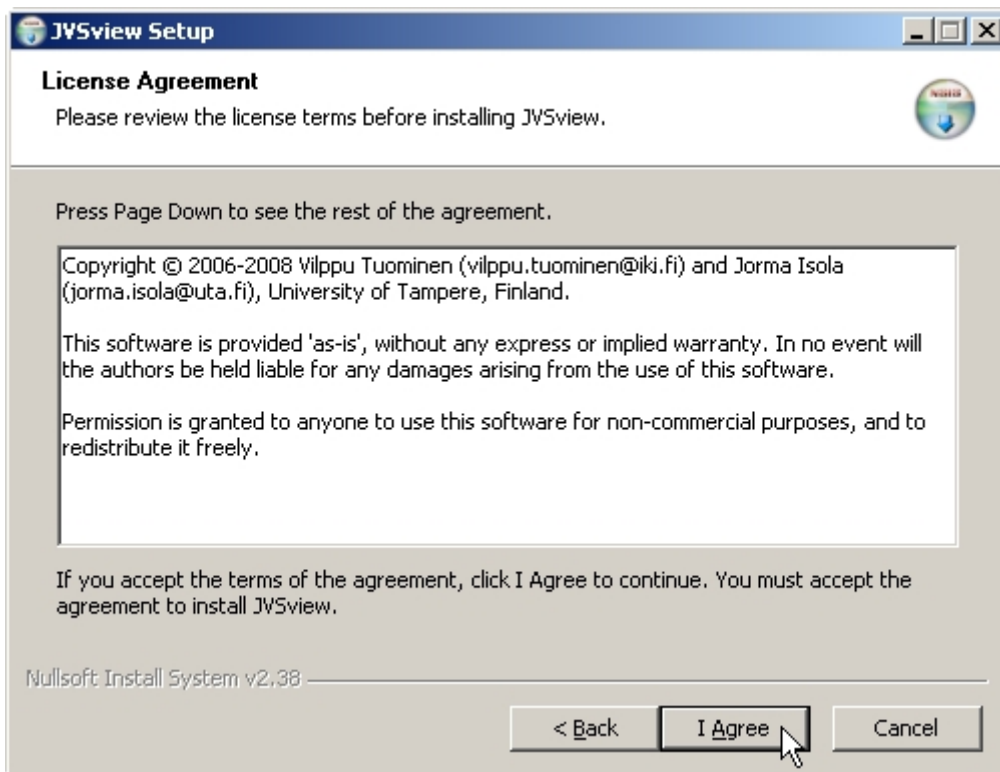
KUVA 2: JVSVIEW:N LATAAMINEN

Asenna ajamalla jvsview-1.2-full.exe -tiedosto. (Kuva 3)

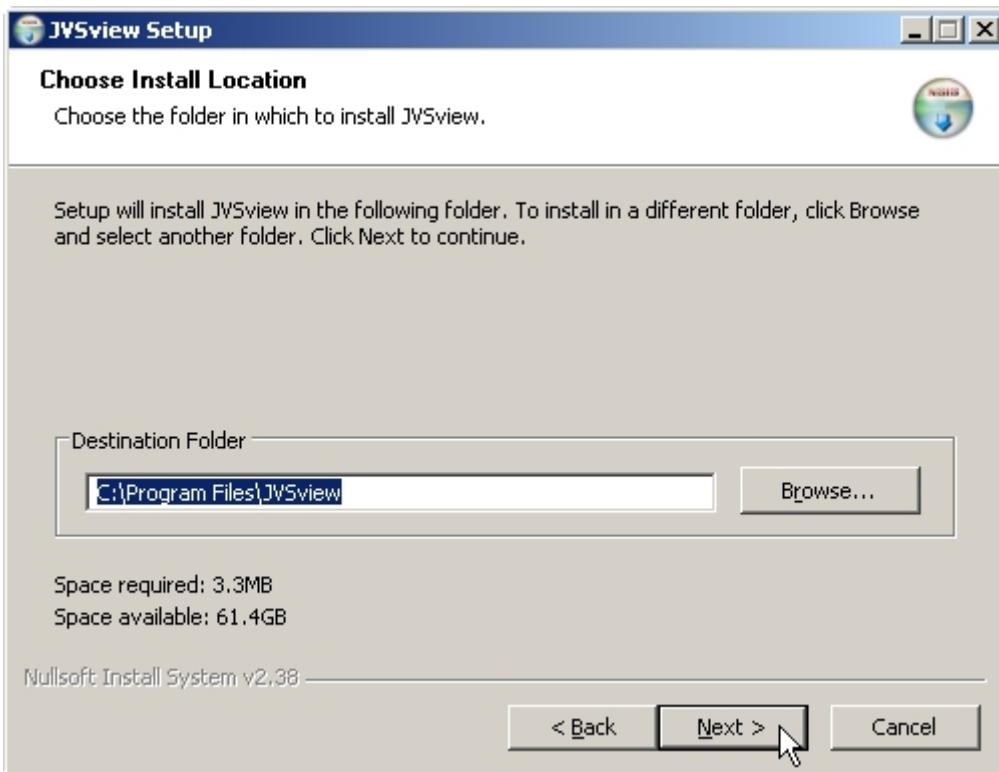


KUVA 3: JVSVIEW ASENNUS

Hyväksy lisenssi ja asenna ohjelma oletussijaintiin. (Kuva 4–Kuva 5)

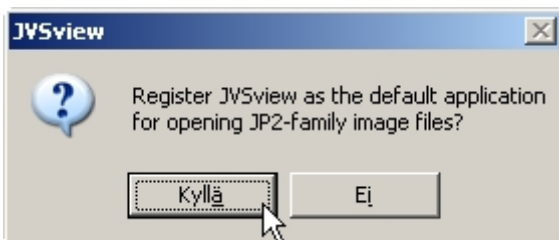


KUVA 4: HYVÄKSY LISENSSI



KUVA 5: OLETUSSIJAINTI

Käynnistä JVSview. Ohjelma ehdottaa itseään oletusohjelmaksi ".jp2" eli JPEG2000-tiedostoille. Hyväksy tämä. (Kuva 6)

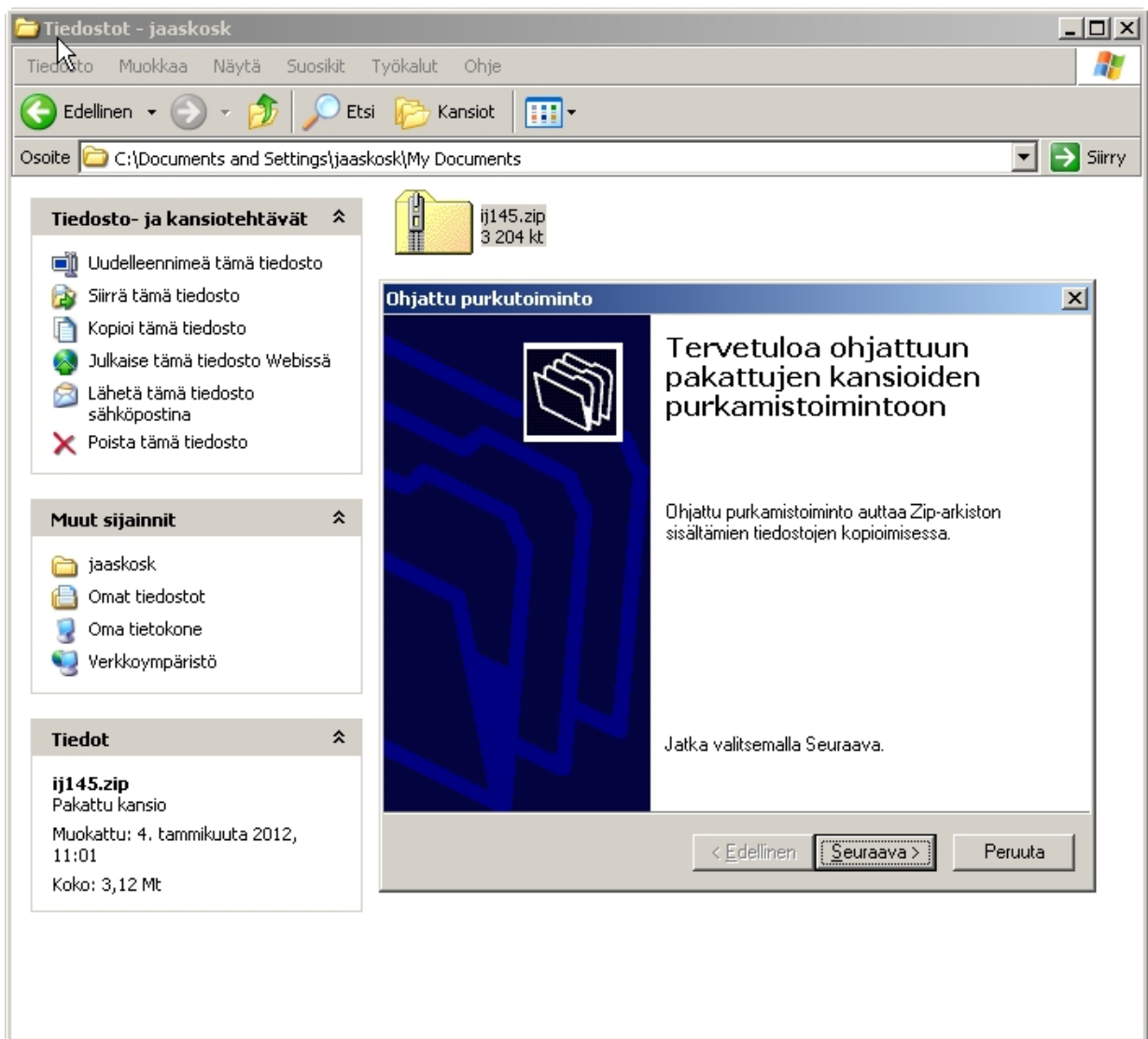


KUVA 6: JPEG2000 OLETUSOHJELMAKSI REKISTERÖINTI

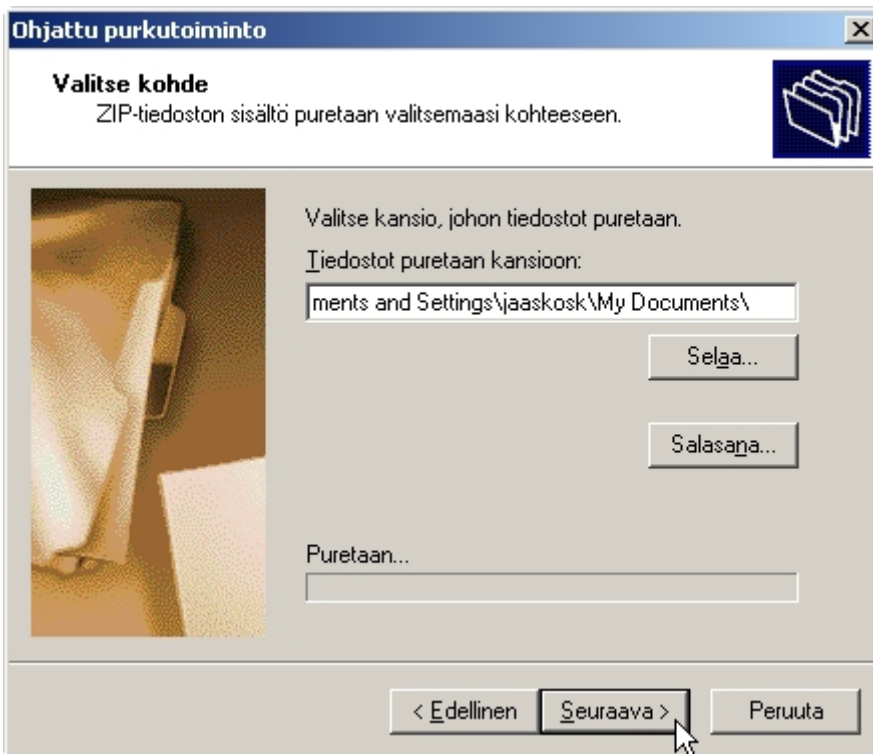
6.2.2 IMAGEJ

Lataa ImageJ osoitteesta <http://imagej.nih.gov/ij/download.html>. Jos sinulla on järjestelmänvalvojan oikeudet, lataa asennuspaketti Windows -> "bundled with 32-bit Java 1.6.0_20". Asenna ImageJ "Omat tiedostot" -kansioon.

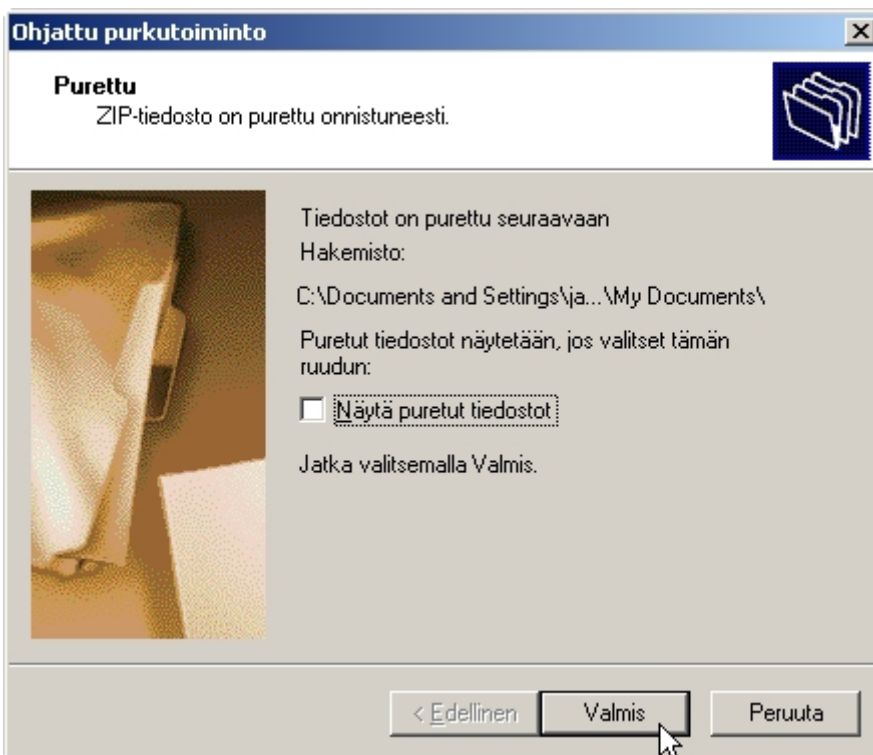
Jos järjestelmänvalvojan oikeuksia ei ole, lataa paketti Platform Independent -> ij145.zip. Pura paketti "Omat tiedostot" -kansioon. (Kuva 7–10)



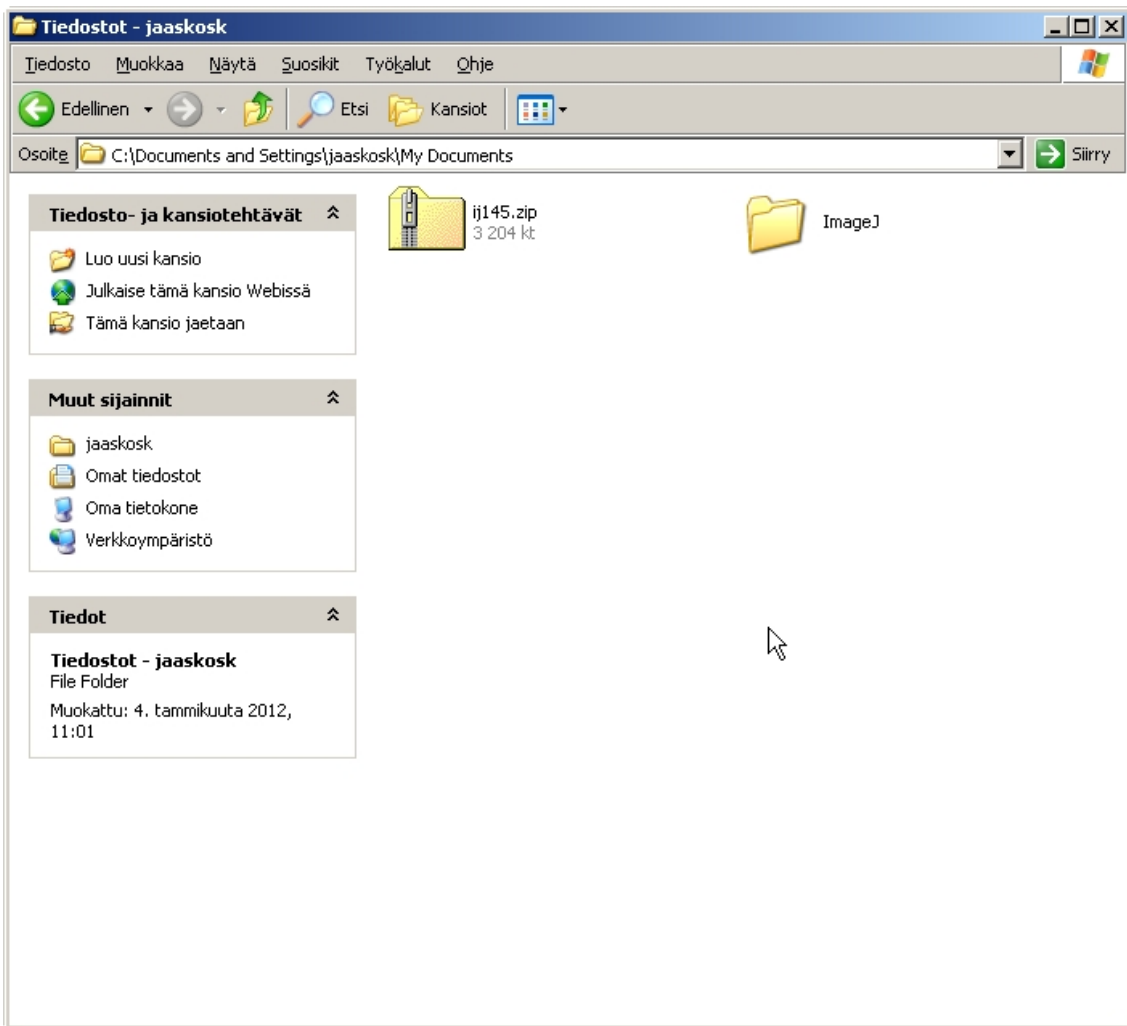
KUVA 7: IMAGEJ ASENNUSPAKETIN PURKAMINEN



KUVA 8: IMAGEJ ASENNUSPAKETIN PURKAMINEN

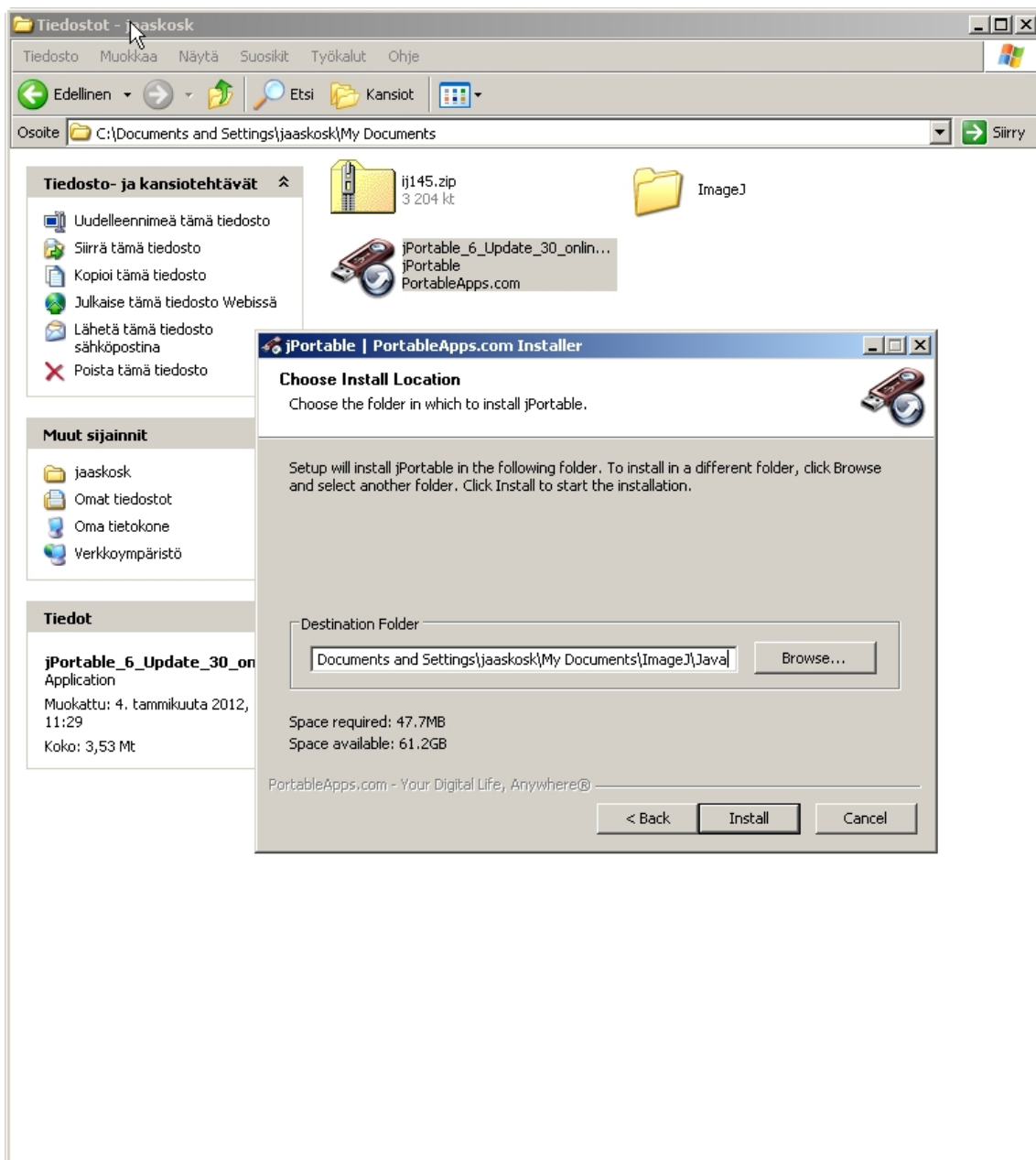


KUVA 9: IMAGEJ ASENNUSPAKETIN PURKAMINEN



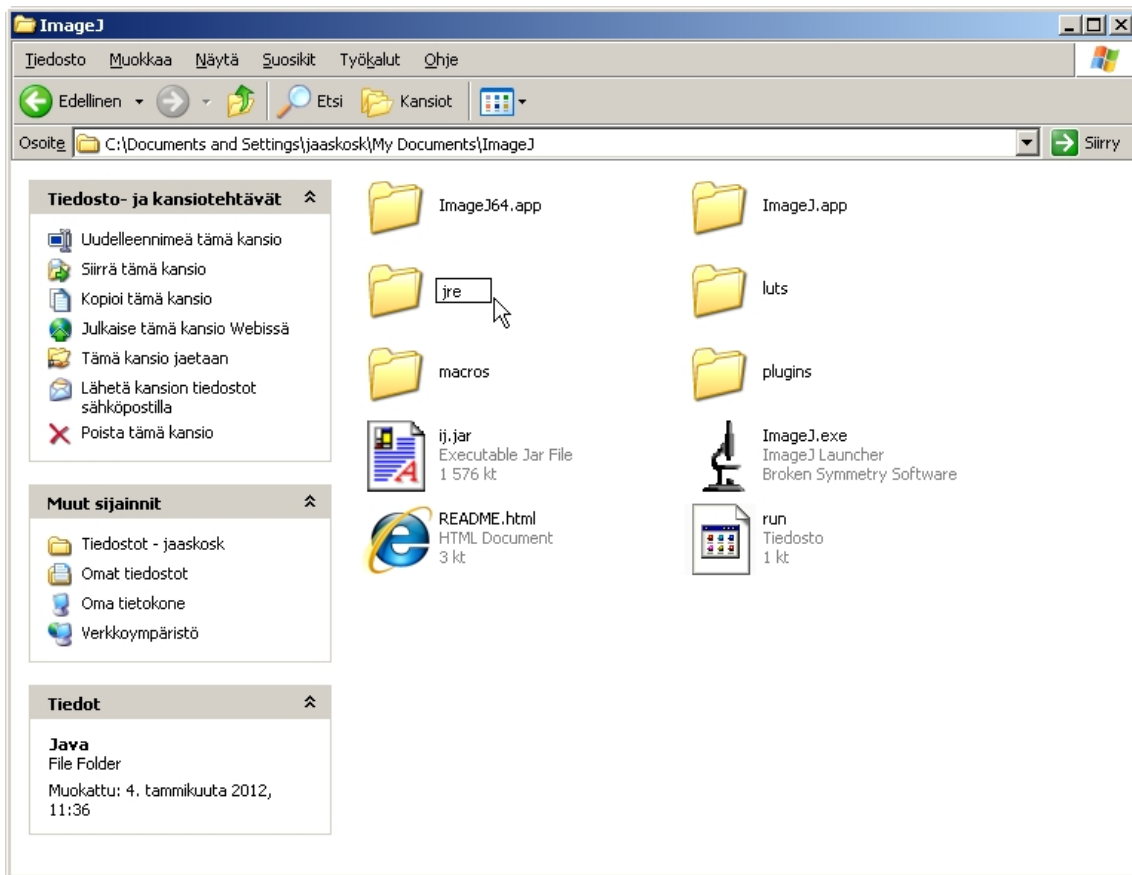
KUVA 10: IMAGEJ ASENNUSPAKETIN PURKAMINEN

Jos latsit ij145.zip-paketin, ja koneella ei ole oikein asennettua javaa, ImageJ ei vielä toimi. Lataa Portable Java osoitteesta http://portableapps.com/apps/utilities/java_portable. Käynnistä asennusohjelma. Hyväksy lisenssi. Kun asennusohjelma kysyy asennuskansiota, selaa äsken purkamaasi ImageJ-kansioon. Asennusohjelma luo sinne Java-alikansion. (Kuva 11)



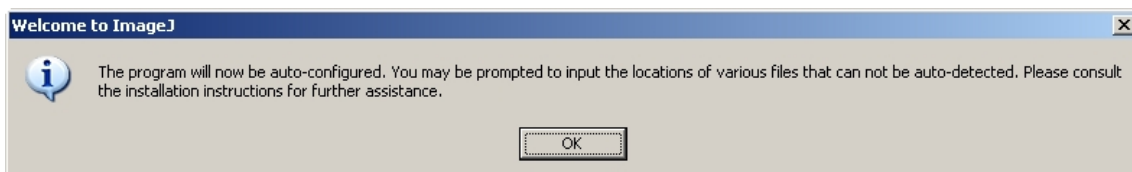
KUVA 11: PORTABLE JAVAN ASENNUS

Suorita asennus loppuun. Mene ImageJ-kansioon ja nimeä uudelleen Java-kansio "jre":ksi.
(Kuva 12)

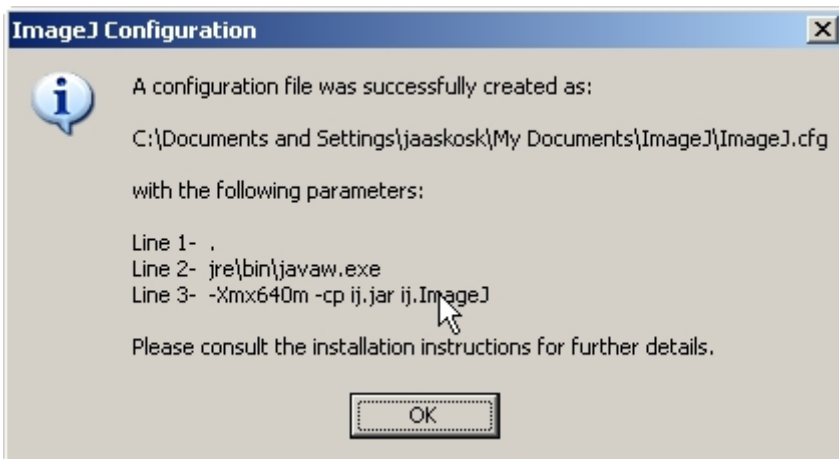


KUVA 12: JAVA -KANSION UUDELLEENNIMEÄMINEN

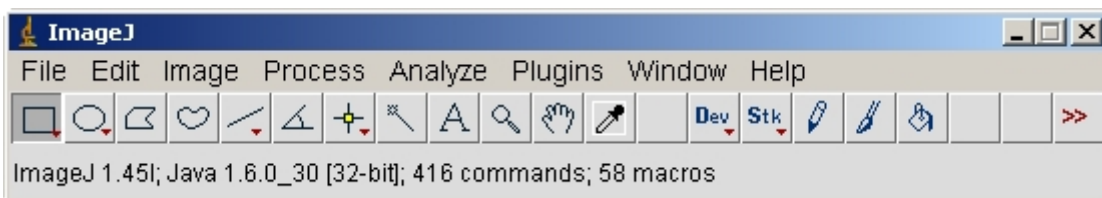
Käynnistä ImageJ.exe. Ohjelma konfiguroi itsensä ja sen pitäisi löytää äsken asennettu java automaattisesti (kuva). Mikäli näin ei käy, täytyy ohjelmalle näyttää javaw.exe:n sijainti, joka löytyy ImageJ\jre\bin –kansioista (Kuvat 13–15).



KUVA 13: IMAGEJ:N AUTOMAATTINEN KONFIGURAATIO

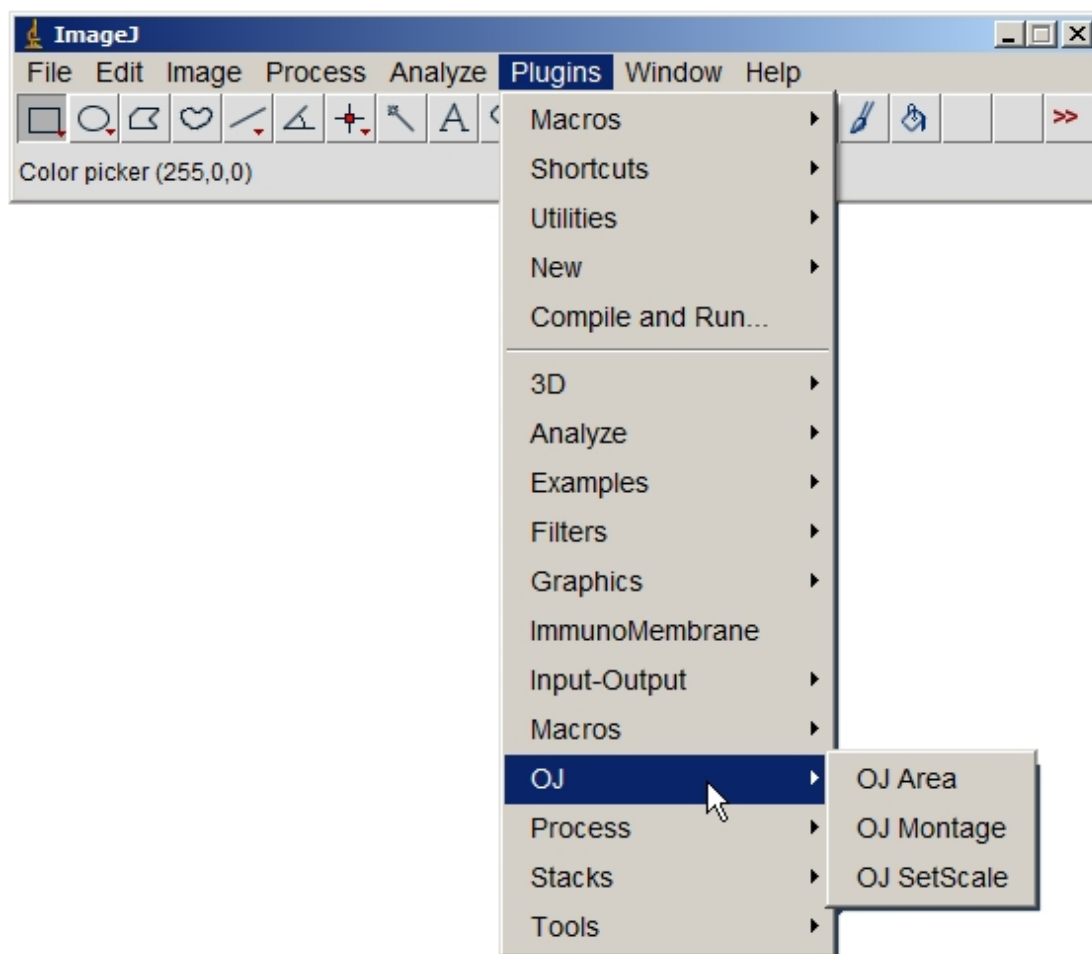


KUVA 14: IMAGEJ:N AUTOMAATTINEN KONFIGURAATIO



KUVA 15: IMAGEJ PÄÄIKKUNA

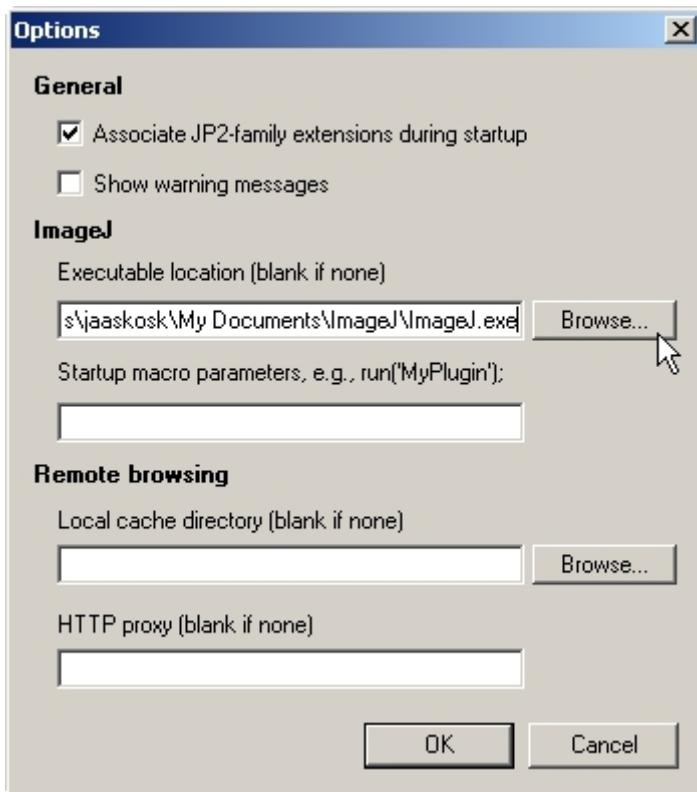
Tee ImageJ\plugins –kansiolle alikansio OJ ja kopioi sinne tiedostot OJ_Area.ijm, OJ_SetScale.ijm ja OJ_Montage.ijm. Käynnistä ImageJ uudestaan. Makrot tulevat näkyviin Plugins>OJ –valikon alle. (Kuva 16)



KUVA 16: PLUGINS-VALIKKO

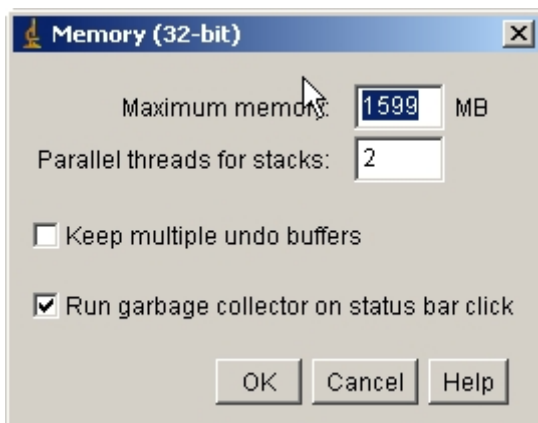
6.2.3 OHJELMIEN KONFIGUROINTI

JVSview:lle täytyy kertoa, missä ImageJ sijaitsee, jotta siirto ohjelmien välillä toimisi. Tämä tapahtuu JVSview:n valikosta Tools>Options kohdassa ImageJ Executable location. Selaa tähän kohtaan ImageJ:n asennuskansiota ImageJ.exe. Muihin kohtiin ei tule muutoksia. (Kuva 17)



KUVA 17: JVSVIEW KONFIGUROIINTI

Koska käsiteltävät kuvat ovat niin suuria, ImageJ:n oletuksena varaama 640 MB muistia ei todennäköisesti riitä. 32-bittisessä Windowsissa ImageJ ei anna asettaa muistivarausta suuremmaksi kuin 1599 MB, johtuen java virtuaalikoneen asettamista rajoituksista. Aseta muistivaraus valikosta Edit>Options>Memory. Varausta ei kannata asettaa suuremmaksi kuin n. 75% tietokoneeseen asennetusta muistista. (Kuva 18)



KUVA 18: IMAGEJ:N MUISTIVARAUS

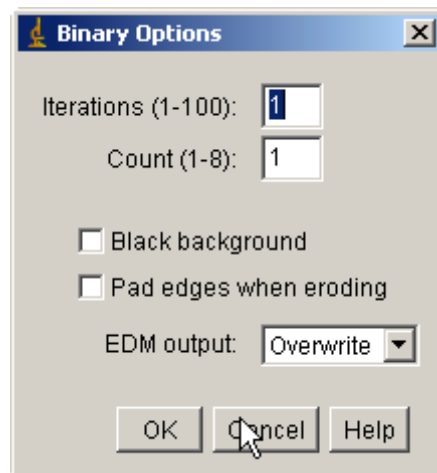
ImageJ:n oletusvärit kannattaa asettaa valikosta Edit>Options>Colors niin, että tausta on valkoinen, piirtoväri musta ja valinnat erottuvat paremmin punaisena. (Kuva 19)

Tarkista vielä, että Process>Binary>Options valikossa Black background ei ole valittuna. (Kuva 20)

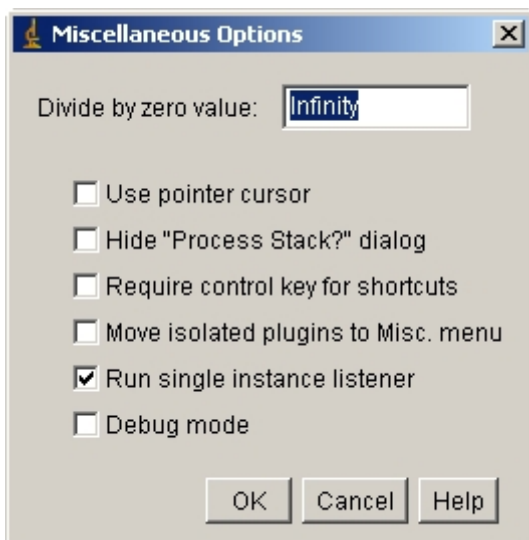
Varmista Edit>Options>Misc –valikosta, että asetus "Run single instance listener" on päällä. (Kuva 21)



KUVA 19: IMAGEJ:N OLETUSVÄRIT



KUVA 20: BINARY OPTIONS

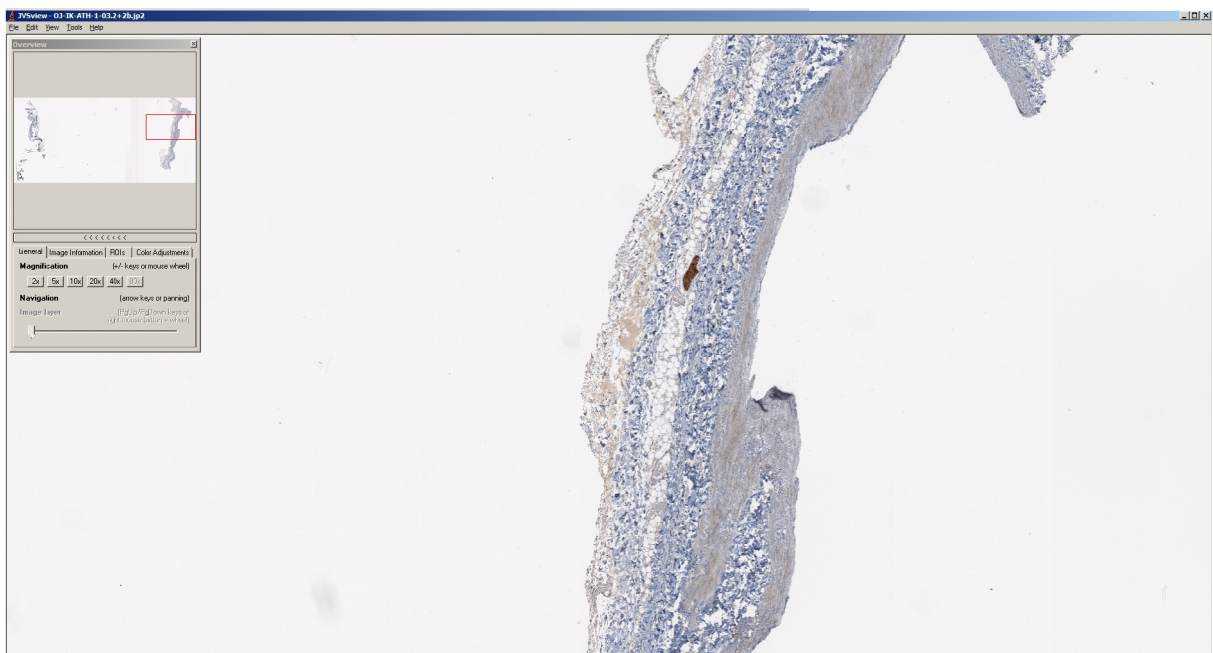


KUVA 21: "SINGLE INSTANCE LISTENER"

6.3 VIRTUAALILASIN KATSELU

Skannerilta virtuaalinäytelasit tulevat JPEG2000-muodossa. Tiedostopääte on ".jp2".

Pakattunakin virtuaalinäytelasien sisältämä tietomäärä on valtava, eikä mahdu tavallisen kuvankäsittelyohjelman muistiin. Näytelasien katseluun käytetään JVSView-ohjelmaa (Tuominen 2008), joka lukee tiedostosta muistiin vain näytössä näkyvän osan kerrallaan. Heti skannauksen jälkeen virtuaalinäytelasit on syytä tarkistaa visuaalisesti skannausvirheiden ja muiden artefaktoiden varalta. Tuplaklikkaamalla virtuaalinäytelasin tiedostoa se aukeaa JVSView-ohjelmaan. (Kuva 22)



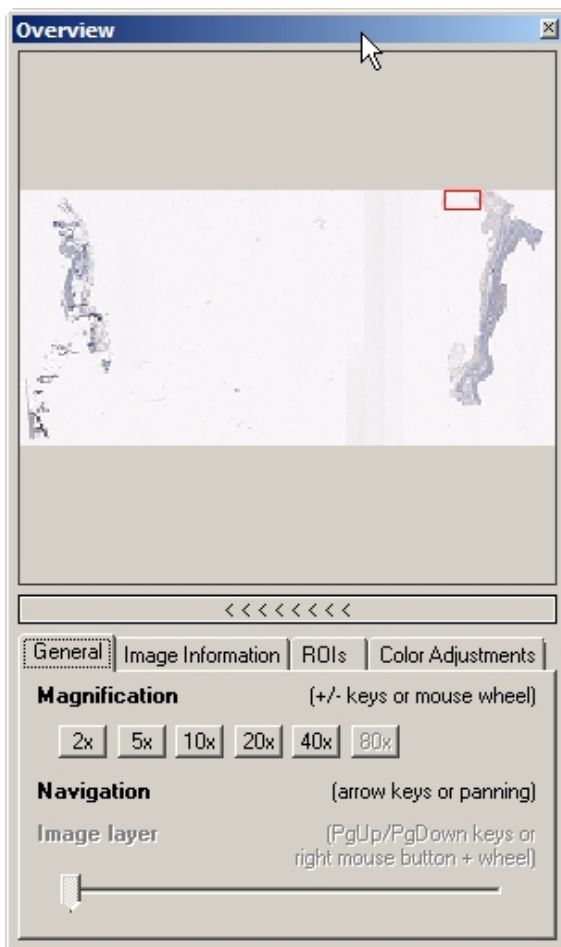
KUVA 22: JVSVIEW

Overview-ruudussa näkyy yleiskuva koko virtuaalinäytelasista. Punainen suorakulmio näyttää pääruudussa näkyvän osan sijainnin. Pääruudussa näkyvää kuvaa voi suurentaa ja pienentää hiiren rullalla, painamalla "+" ja "-" -näppäimiä tai klikkaamalla Overview-ruudun suurennusvalitsinta ("Magnification", esimerkiksi "20X"). Näytelasia voi liikuttaa raahaamalla hiirellä joko Overview-ruudun punaista suorakulmiota tai raahaamalla pääruudussa näkyvää kuvaa. Kätevin tapa on kuitenkin käyttää nuolinäppäimiä. Nuolinäppäintä normaalisti painaessa kuva siirtyy vain vähän kerrallaan, mutta pitämällä SHIFT-näppäin pohjassa kuva siirtyy aina täsmälleen yhden näkyvissä olevan ruudun verran. Tätä ominaisuutta käytetään hyväksi siirrettäessä kuvia ImageJ-ohjelmaan. Kannattaa huomata, että kuva-alueen reunan tullessa vastaan siirtymä ei aina olekaan koko ruudun verran.

Käytössä oleva Aperio ScanScope® XT –skanneri ja sen ohjelmisto on kalibroitu niin, että JVSView-ohjelman suurennoksella ”40X” näytöllä näkyvä pikseli vastaa 0,5 mikrometriä näytelasilla. Vastaavasti ”20X” suurennuksella yksi näytöllä näkyvä pikseli vastaa yhtä mikrometriä fyysisellä näytelasilla. Tämän mittasuhteen paikkansapitävyys on täysin riippuvainen skannerin huolellisesta kalibroinnista ja oikeasta skannaustekniikasta. Esimerkiksi tarkennustason merkittävästi muuttuessa muuttuu myös optinen suurennus.

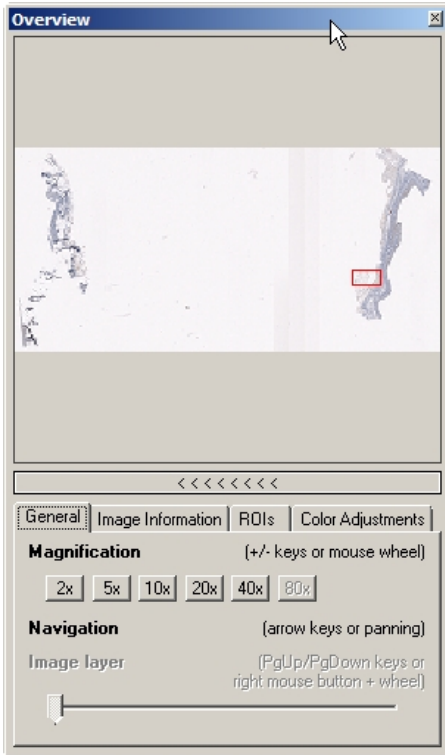
6.4 KUVIEN SIIRTO IMAGEJ-OHJELMAAN

Kuvien siirrossa käytetään edellä mainittua 1 pikseli = 1 mikrometri –mittasuhdetta. Valitse JVSView-ohjelmassa tätä varten ”20X”-suurennus. Koska tällä suurennuksella koko näyte ei (todennäköisesti) mahdu kerralla näkyviin, on kuva siirrettävä useassa osassa. Hahmota suorakulmion muotoinen alue, joka sisältää kyseessä olevan näytteen. Siirry nuolinäppäimillä tuon alueen vasempaan yläkulmaan. (Kuva 23)

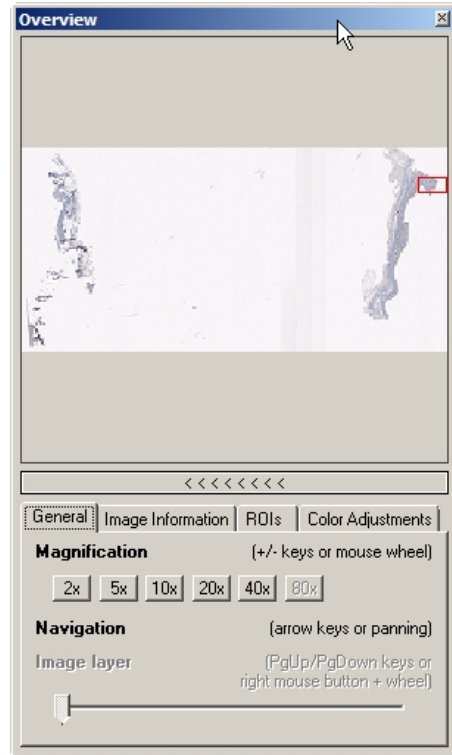


KUVA 23: VASEN YLÄKULMA

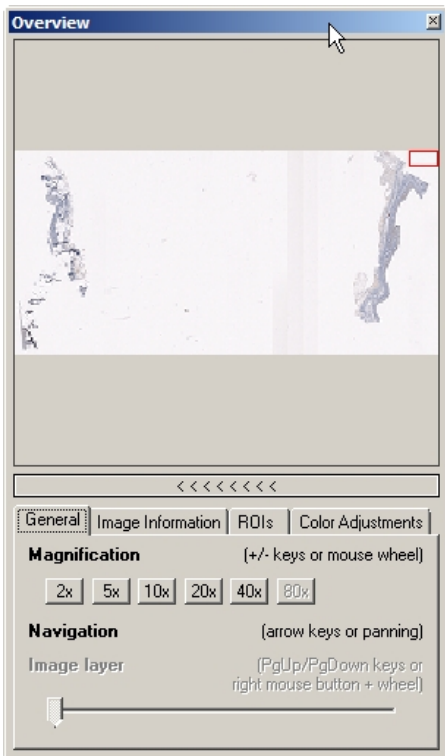
Pidä SHIFT-näppäin pohjassa ja siirry nuolinäppäimillä alaspäin, varmistaen, että koko näytteen vasen reuna mahtuu alueeseen (Kuva 24). Tee samoin ylä-, ala-, ja oikealle reunalle (Kuvat 25–27).



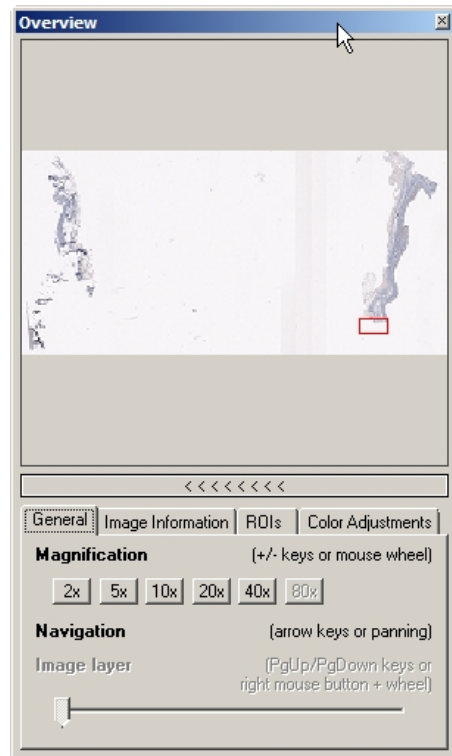
KUVA 24: VASEN REUNA



KUVA 26: OIKEA REUNA

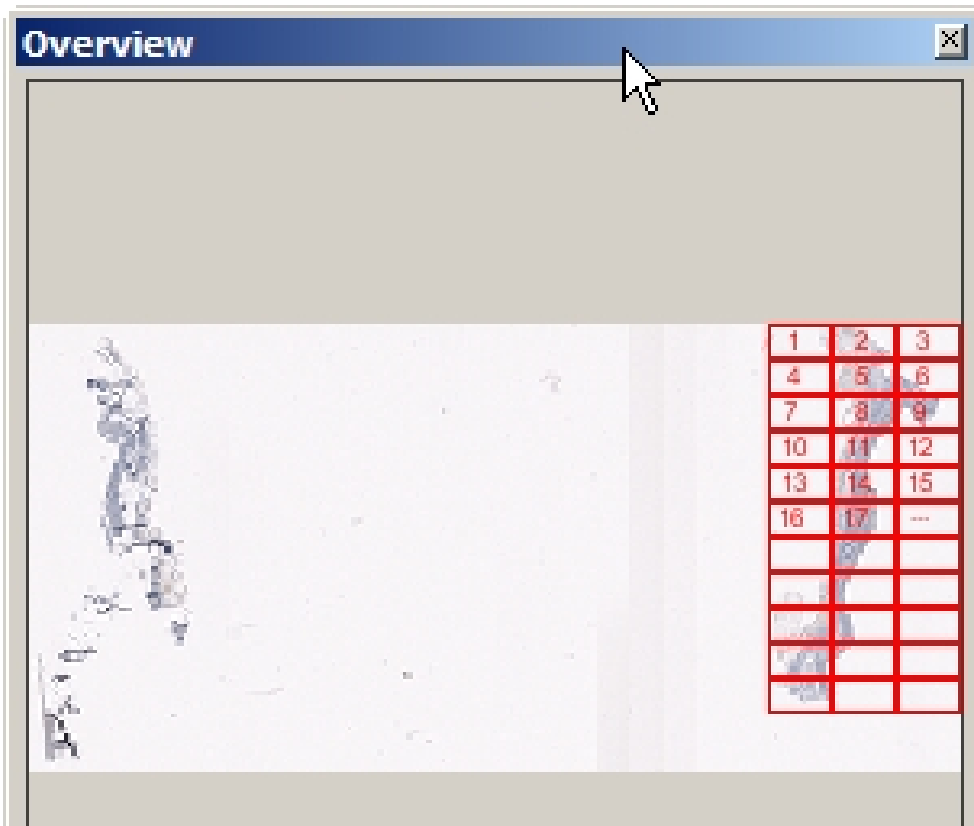


KUVA 25: OIKEA YLÄKULMA



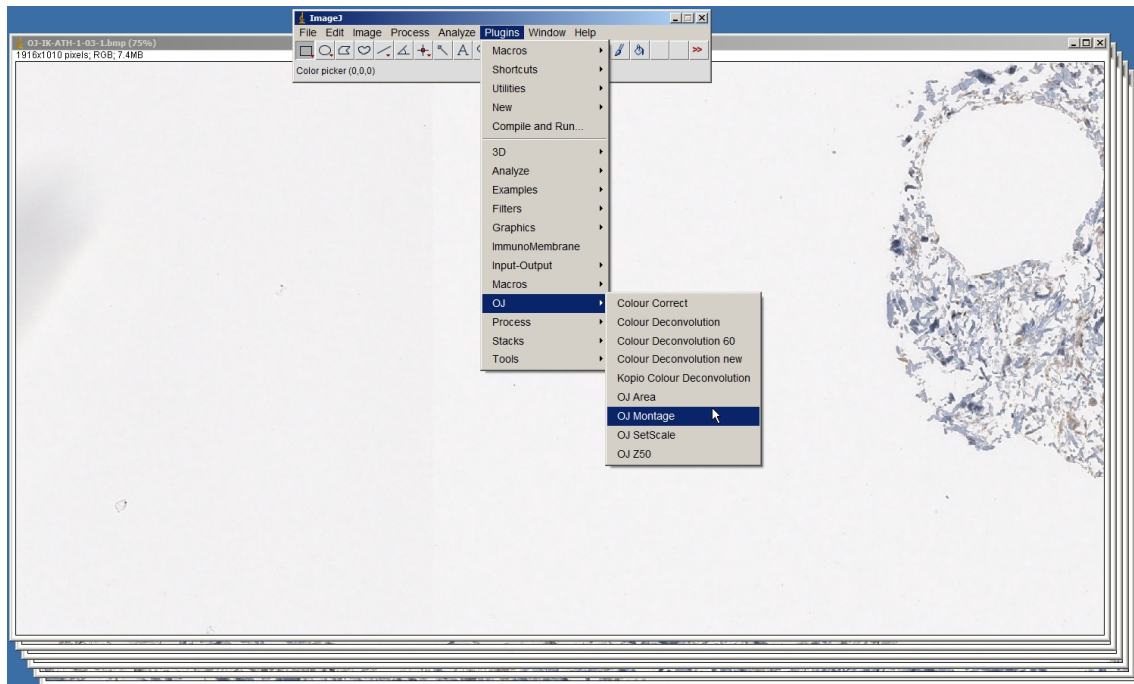
KUVA 27: ALAREUNA

Avaa ImageJ-ohjelma, ja siirry takaisin JVSView –ohjelmaan. Siirry SHIFT-näppäin pohjassa takaisin vasempaan yläkulmaan. Paina CTRL+A. Näkyvä osa virtuaalinäytelasia kopioituu ImageJ-ohjelmaan. Jatka kopiointia rivi kerrallaan aina vasemmalta oikealle. Huomaa, että koko alueen jokainen ruutu on kopioitava, vaikka kyseisessä ruudussa ei olisikaan näkyvissä näytettä. Muuten ImageJ ei osaa koota kopioituja ruutuja kokonaisuudeksi. (Kuva 28)



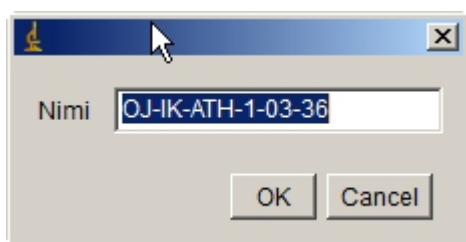
KUVA 28: RUUDUT ON KOPIOITAVA OIKEASSA JÄRJESTYKSESSÄ

Nyt ImageJ-ohjelmassa on useita erillisiä kuvia auki. Seuraavaksi kuvat yhdistetään OJ_Montage –makrolla. Voit sulkea JVSView –ohjelman. Valitse ImageJ:n valikosta Plugins -> OJ -> OJ Montage. (Kuva 29)



KUVA 29: KÄYNNISTÄ OJ-MONTAGE

Makro tarjoaa ensin mahdollisuutta nimetä käsiteltävä kuva. Makro ehdottaa valmiiksi virtuaalinäytelasin tiedostonimestä muodostettua nimeä. Näytelasit on jo skannatessa hyvä nimetä näytenumeron ja värjäyksen mukaan (kuva 30). Seuraavaksi kysytään sarakkeiden ja rivien määrää (kuva 31).



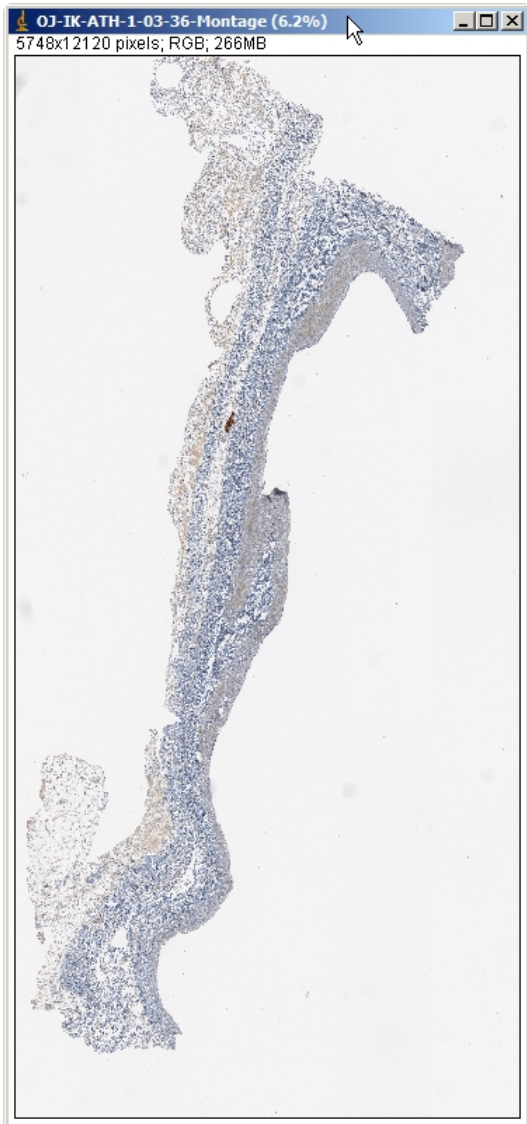
KUVA 30: NIMEÄ KUVA



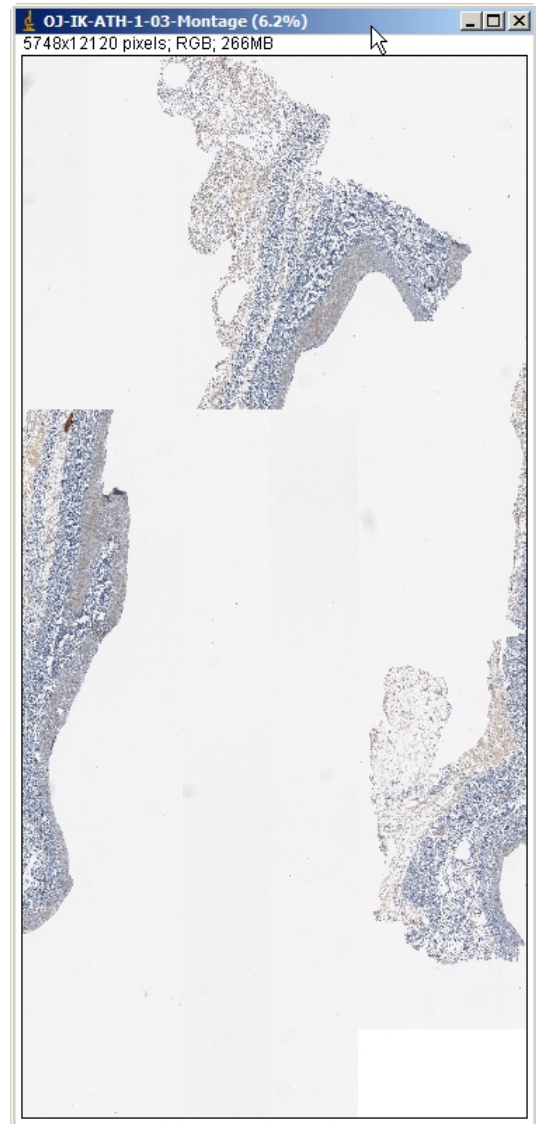
KUVA 31: SYÖTÄ SARAKKEET JA RIVIT

Syötettyäsi oikeat arvot, makro muodostaa ensin kuvista "pinon" (image stack), ja sen jälkeen montaasin (montage). Tarkista, että kuva on kunnossa (kuva 32). Väärässä

järjestyksessä siirretyt kuvat, tai yksittäisten kuvien puuttuminen välistä näkyy montaaissa helposti (kuva 33). Nyt voit sulkea ylimääräiset ikkunat.



KUVA 32: OIKEIN RAKENNETTU MONTAASI

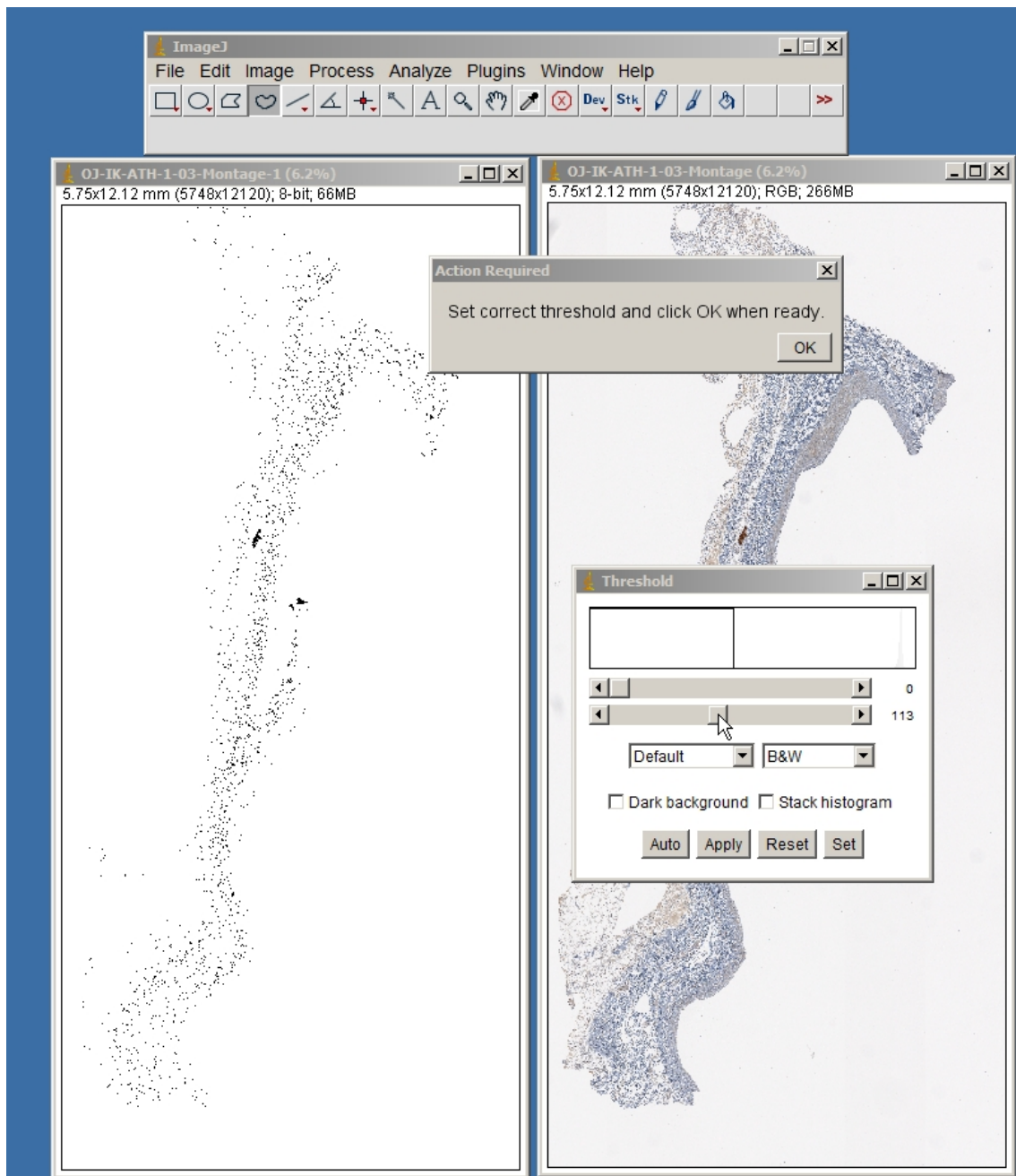


KUVA 33: VÄÄRIN RAKENNETTU MONTAASI

6.5 PINTA-ALAN MITTAAMINEN

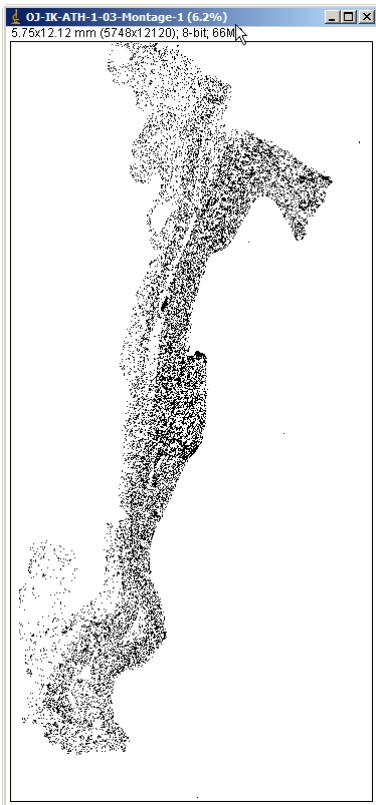
Pinta-alan mittaaminen tehdään montaasikuvasta. Valitse aktiiviseksi ikkunaksi montaasikuva, sen jälkeen valikosta Plugins -> OJ -> OJ Area. Ensimmäisenä makro kysyy suurennoksen, jota JVSView -ohjelmassa käytettiin. Oletuksena on 20x -suurennos. Tämän perusteella kuvalle asetetaan mittakaava, jotta mittaustulokset saadaan oikeissa yksiköissä. Seuraavaksi makro tekee kuvasta kopion ja kysyy sille nimeä. Voit hyväksyä oletuksen. Kopio

muunnetaan ensin harmaasävykuvaksi ja sen jälkeen raja-arvoistetaan (thresholding). Raja-arvoistuksella kuva muutetaan kaksiväriseksi. Raja-arvoa vaaleammat pikselit muutetaan puhtaan valkoisiksi ja sitä tummemmat pikselit puhtaan mustiksi. Mustat pikselit edustavat näytettä, valkoiset taustaa.

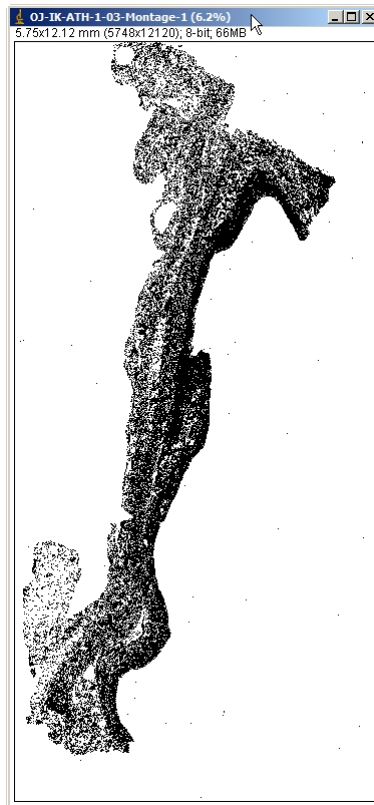


KUVA 34: THRESHOLDING

Siirrä ”Threshold” –ikkunan alemmaa säädintä kunnes mahdollisimman suuri osa näytteen alueesta on mustaa, mutta niin, ettei taustan alueelle tule juurikaan mustia pikseleitä (kuva 34). Näytteen alueella on luonnostaan valkoisiksi jääviä kohtia, mm. näytteessä olevat rasvasolut, joiden sisältö histologisessa prosessoinnissa poistuu. Näillä ei ole merkitystä, algoritmi täyttää ne myöhemmin. Samoin taustan alueelle jää väkisinkin yksittäisiä mustia pikseleitä. Näilläkään ei ole merkitystä, kunhan ne eivät ole muutamaa pikseliä suurempia. Alla olevissa kuvissa on esimerkkejä raja-arvoistuksesta. Vasemmanpuoleinen on turhan vaalea, keskimmäinen sopiva ja oikeanpuoleinen selvästi liian tumma (kuvat 35–37). Kun olet valinnut sopivan raja-arvon, klikkaa Action required –ikkunassa ”OK”. Threshold –ikkunan voit nyt sulkea.



KUVA 35: LIIAN VAALEA



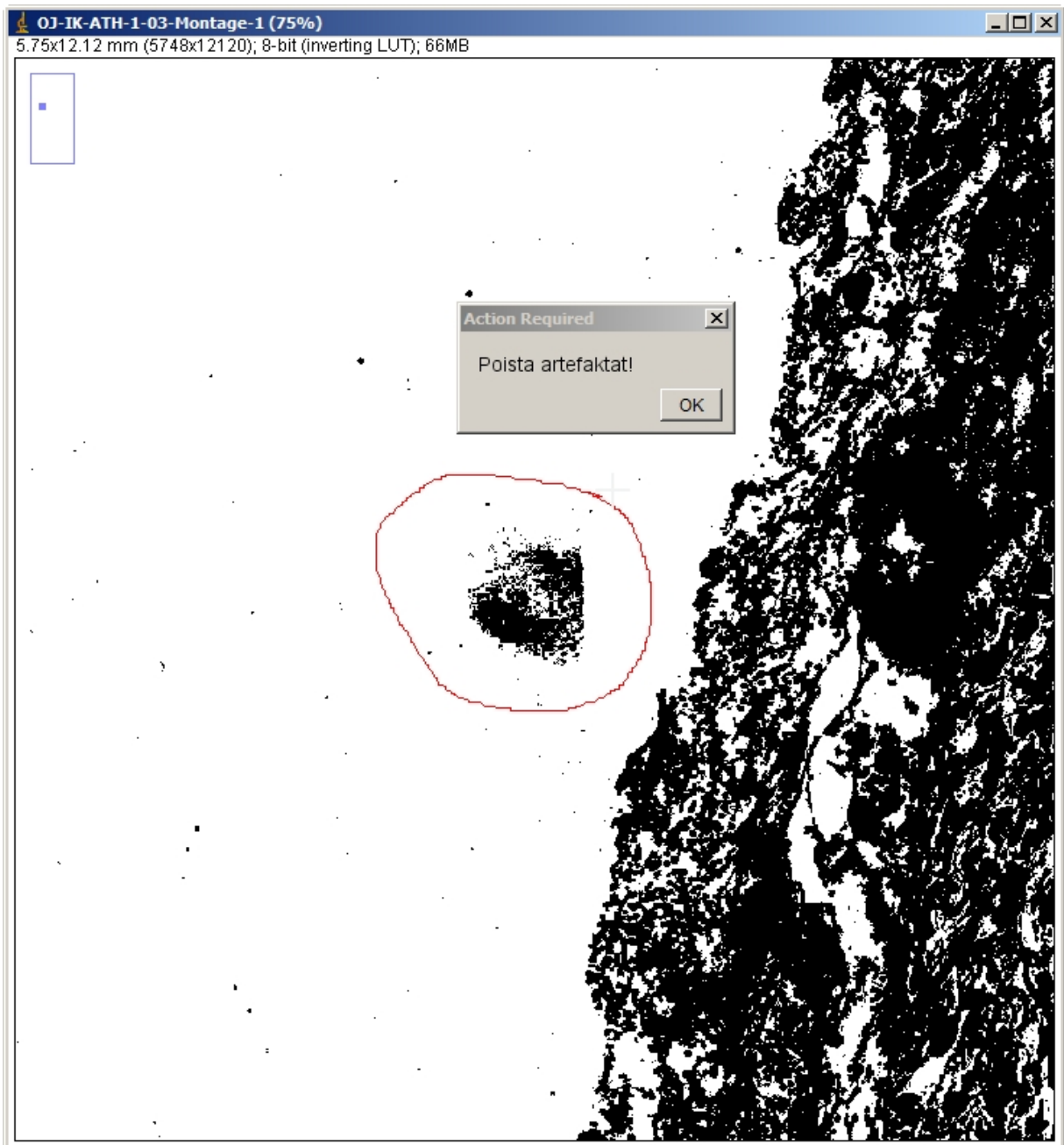
KUVA 36: SOPIVA



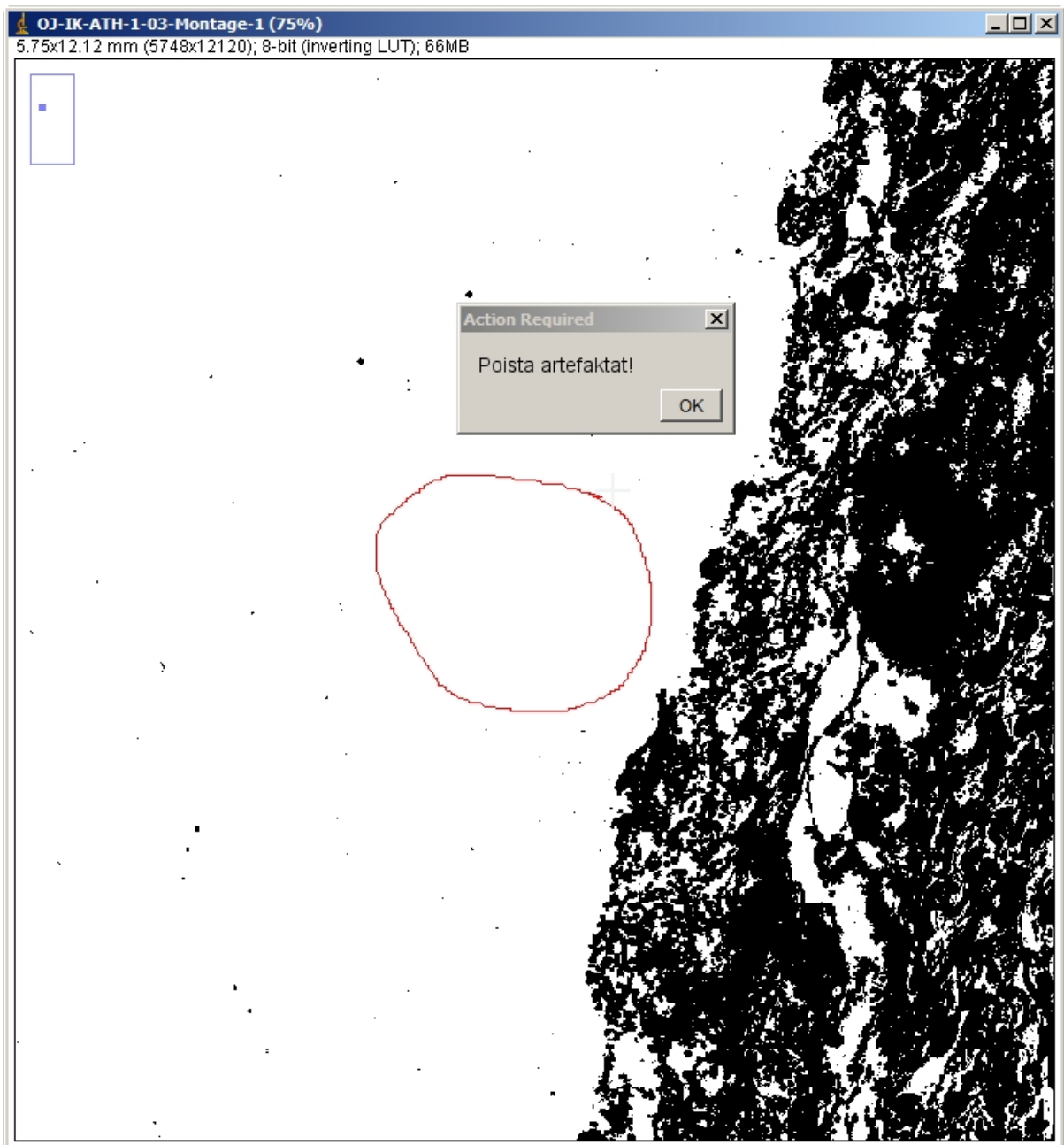
KUVA 37: LIIAN TUMMA

Seuraavaksi makro tarjoaa mahdollisuutta poistaa taustaan jääneitä artefaktoja. Useimmissa tapauksissa tätä toimintoa ei tarvita jos näytteiden käsittely ja skannaus on tehty huolella. Joskus kuitenkin taustan (tai näytteenkin) alueella on esimerkiksi ilmakuplia, roskia tai vaurioita peitelasissa. Algoritmi tulkitsee nämä virheellisesti näytteen pinta-alaan kuuluviksi.

Artefaktojen poistamiseksi algoritmi valitsee automaattisesti käyttöön Freehand selection – työkalun. Numeronäppäimistön + ja -näppäimillä voit suurentaa ja pienentää kuvaa. Pitämällä välilyöntinäppäimen pohjassa, hiiren kuvake muuttuu kädeksi, jolla voit raahata kuvaa edestakaisin. Saatuaasi artefakta-alueen hyvin näkökenttään, ympyröi se valintatyökalulla (kuva 38) ja paina CTRL+X, jolloin valinta muuttuu valkoiseksi (kuva 39). Voit jatkaa samalla tavalla muiden artefaktojen poistamista. Kun olet valmis, klikkaa "OK".



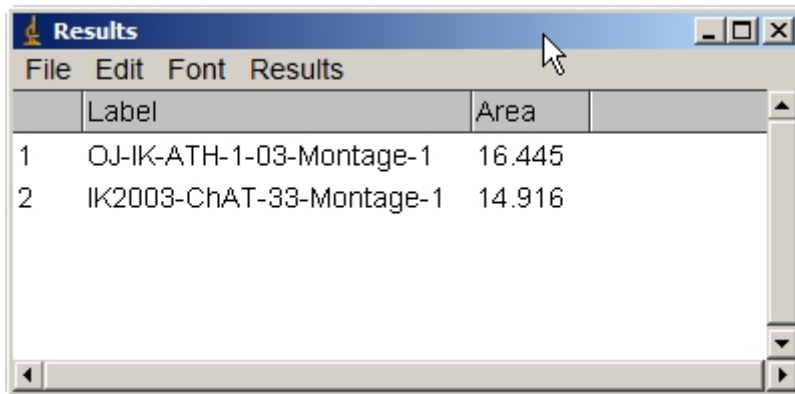
KUVA 38: YMPÄRÖI ARTEFAKTA



KUVA 39: PAINA CTRL+X

Seuraavaksi algoritmi täyttää näytteen alueelle jääneet reiät. Tässä kestää tietokoneen tehosta riippuen hetken aikaa. Älä klikkaile muita ikkunoita algoritmin suorituksen aikana. ImageJ kohdistaa toimintonsa aina aktiivisena olevaan ikkunaan. Kun algoritmi on valmis, aukeaa Results -ikkuna. Tarkista, että algoritmi on värjännyt koko näytteen alueen mustaksi, eikä taustaan ole ilmestynyt ylimääräisiä mustia alueita. Tarvittaessa palaa alkuperäiseen montaasi-ikkunaan ja aloita pinta-ala-algoritmi uudestaan.

Results –ikkunaan tulee jokaista mittausta kohti yksi rivi, jossa on mitatun ikkunan nimi, sekä saatu pinta-ala neliömillimetreissä. Esimerkissä on mitattu jo kaksi näytettä, OJ-IK-ATH-1-03 –näytteen pinta-ala on 16,445 neliömillimetriä ja IK2003-ChAT-33 –näytteen 14,916 neliömillimetriä (kuva 40). Results –ikkunan voi jättää auki seuraavia mittauksia varten, mutta muut kuva-ikkunat on syytä sulkea, koska ne sotkisivat montaasin teon.



The screenshot shows a software window titled 'Results' with a menu bar containing 'File', 'Edit', 'Font', and 'Results'. The window contains a table with two columns: 'Label' and 'Area'. There are two rows of data. A mouse cursor is pointing at the 'Results' menu item.

	Label	Area
1	OJ-IK-ATH-1-03-Montage-1	16.445
2	IK2003-ChAT-33-Montage-1	14.916

KUVA 40: NÄYTTEIDEN PINTA-ALAT

7 POHDINTA

Tavoitteena oli kehittää histologisen kudoksenäytteen pinta-alan mittaava algoritmi. Tässä onnistuttiin hyvin. Menetelmän käyttökelpoisuutta ja luotettavuutta arvioitaessa on otettava huomioon molemmat vaiheet: näytelasien digitoiminen ja digitaalista kuvatieta käsittelevä algoritmi.

Onnistunut näytelasien digitoiminen edellyttää monta olennaista työvaihetta. Ensimmäinen vaihe on histologisen näyttelasin normaali valmistusprosessi. Näytelasien on oltava riittävän tasalaatuisia, jotta niitä voidaan analysoida automatisoidusti. Ihminen on näytelasia katsoessaan paljon kykenevämpi mukautumaan laatuvariaatioihin kuin tietokonealgoritmi. Tämä edellyttää automaattiseen analysointiin tulevilta näytelaseilta jopa parempaa tasalaatuisuutta kuin perinteisessä patologian laboratoriossa. Toki nykyaikaisen patologian laboratorion laatu on diagnostisten näytteiden ja diagnostiikkaan käytettävien värjäysten kohdalla hyvin standardoitu. Tilanne saattaa olla toinen esimerkiksi tutkimuskäytössä olevien värjäysten tai näyteprosessointien suhteen. Pienillä näytemäärillä on hankalampi varmistaa tasaista laatua.

Seuraava vaihe on näytelasien kuvaaminen eli digitoiminen. Skannauksessa käytettävä optinen suurennos on samaa suuruusluokkaa perinteisen mikroskoopin kanssa. Tästä seuraa hyvin kapea syväterävyysalue. Vaikka nykyaikaiset skannausalgoritmit löytävät tehokkaasti oikean tarkennustason, saattaa tässä silti tulla virheitä, jolloin osa näytteen alueesta jää epätarkaksi. Lisäksi tarkennustason merkittävä muuttuminen muuttaa myös optista suurennosta, jolloin kuvan mittakaava muuttuu. Tällainen tilanne olisi esimerkiksi se, että näytelasit on laitettu väärinpäin skanneriin. Mittakaavan muuttuessa ei algoritmin antama pinta-ala neliömillimetreissä luonnollisesti vastaa todellisuutta, ellei mittakaavaa muuteta myös algoritmissa. Toisaalta mittakaavan pienellä muuttumisella ei välttämättä ole merkitystä, mikäli kaikki muutkin mittaukset tehdään samasta kuvasta.

Toinen mahdollinen laatuvirhe digitoinnissa on kuvan huono kontrasti. Tähän vaikuttaa osaltaan myös valkotasapainon oikeellisuus. Liian matalakontrastisessa kuvassa vaalean ja tumman pään sävyt ovat liian lähellä toisiaan. Tausta ei ole kunnolla valkoinen vaan jää harmahtavaksi, ja toisaalta näyte ei ole väreiltään riittävän tumma vaan jää hailakaksi. Tällöin taustan ja näytteen erottaminen raja-arvoistamalla muuttuu hankalaksi. Toisaalta liian kontrastikkaassa kuvassa ongelmana on kuvainformaation leikkautuminen sekä vaaleasta että tummasta päästä. Esimerkiksi näyte saattaa kuvautua joiltain osin niin tummana, ettei solustruktuureita enää kuvasta erota.

Itse algoritmin heikkouksia ja vahvuuksia pohdittaessa vahvuuksien puolelle tulee nopeus, tarkkuus ja toistettavuus. Valmistelutyöt vaativat luonnollisesti oman aikansa, mutta digitaalisen patologian kehittyessä skannausmenetelmät nopeutuvat ja automaatio paranee. Algoritmi erottelee kuvasta näytteen alueen muutamassa kymmenessä sekunnissa riippuen käytettävän tietokoneen laskentatehosta. Näytteen ääriviivojen piirtäminen piirtopöytää apuna käyttäen vie jopa 30–45 minuuttia kun se tehdään käsin ja vastaavalla tarkkuudella. Algoritmi myös mittaa pinta-alan samasta kuvasta aina samalla tavalla, jos raja-arvoistus tehdään samalla arvolla.

Heikkoutena on se, ettei algoritmi kykene tulkitsemaan kuvainformaatiota millään tavalla. Näytteen ja taustan erottelu tehdään värjäyksen tummuuteen perustuen, eikä algoritmi osaa erottaa kuvassa tummina näkyviä artefaktoja näytteestä. Ilmakuplat ja erilaiset roskat lasketaan näytteen pinta-alaan, ellei niitä käsin poisteta. Algoritmi ei myöskään erottele epätarkoiksi jääneitä alueita muusta näytteestä.

Toistettavuutta heikentää manuaalinen raja-arvoistus. Erilaisia raja-arvoistus -algoritmeja on olemassa, ja kehitystyön aikana lupaavalta näytti ImageJ:n sisäänrakennettu Intermoden -algoritmi. Tätä ei kuitenkaan ehditty testaamaan riittävän laajalla näytemateriaalilla, jotta sen käyttökelpoisuutta voisi kunnolla arvioida. Ihmisen visuaalisesti tekemä raja-arvoistus on toistaiseksi paras tapa varmistaa, ettei näytteestä puutu osia ja ettei taustaan ilmaannu

liikaa artefaktaa. Toisaalta algoritmi hakeutuu kohti samaa pinta-alaa vaikka raja-arvoistus vaihtelisikin jonkin verran.

Kokonaisuutena menetelmä on nopea ja tarkka tapa mitata kudosnäytteen pinta-ala digitoidusta virtuaalinäytelasista, kunhan pidetään mielessä edellä esitetyt rajoitukset. Luonnollisesti, jos käytetään muita värjäyksiä, skannereita tai ohjelmaversioita kuin edellä on mainittu, täytyy menetelmän käyttökelpoisuus, luotettavuus ja mittakaava määrittää uudestaan.

VIITTEET

Dabbs D. Diagnostic immunohistochemistry. Second edition. Churchill Livingstone 2006

Eteisvärinä. Käypä hoito –suositus. Suomalaisen Lääkäriseuran Duodecimin ja Suomen Kardiologisen Seuran asettama työryhmä. Helsinki: Suomalainen Lääkäriseura Duodecim 2011 [päivitetty 28.1.2011] www.kaypahoito.fi

Jalife J. Déjà vu in the theories of atrial fibrillation dynamics. Cardiovasc Res 2011;89:766-75

Kholová I, Kautzner J. Anatomic characteristic of extension of atrial myocardium into the pulmonary veins in subjects with and without atrial fibrillation. PACE 2003;26:1348-55

Kholová I, Kautzner J. Morphology of Atrial Myocardial Extensions Into Human Caval Veins: A Postmortem Study in Patients With and Without Atrial Fibrillation. Circulation 2004;110:483-8

Koskinen M, Hietaharju A, Kyläniemi M, ym. A quantitative method for the assessment of intraepidermal nerve fibers in small-fiber neuropathy. J Neurol 2005;252:789-94

Laitio M., Vaajalahti P. Tautiopien perusteet. 6. painos. Porvoo: WSOY 1992

Mutterer J., Rasband W. ImageJ Macro Language, Programmer's Reference Guide. http://rsbweb.nih.gov/ij/docs/macro_reference_guide.pdf, 2011

Tuominen V. JPEG2000-standardin hyödyntäminen virtuaalimikroskopiassa. Pro gradu – tutkielma, Tampereen yliopisto, Tietojenkäsittelytieteiden laitos. 2008

Young I.T., Gerbrands J.J., van Vliet L.J., Fundamentals of Image Processing, version 2.3 Delft University of Technology, Delft, 2007

Weinstein, R.S. Innovations in medical imaging and virtual microscopy. Hum Pathol 2005;36:317-9

LIITE 1

Makrojen OJ_Montage.ijm, OJ_SetScale.ijm ja OJ_Area.ijm lähdekoodit.

TAULUKKO 2: OJ_MONTAGE.IJM

```
// OJ_Montage.ijm
// Versio 0.6
// Copyright © 2013 Oskari Jääskeläinen

name=getTitle();
name=getString("Nimi", name);
n= nImages();

run("Images to Stack", "name=&name title=[] use");

Dialog.create("Järjestys");
Dialog.addString("Title:", name);
Dialog.addNumber("Sarakkeita:", n);
Dialog.addNumber("Rivejä:", 1);

Dialog.show();
title = Dialog.getString();
width = Dialog.getNumber();
height = Dialog.getNumber();

run("Make Montage...", "columns=&width rows=&height scale=1 first=1
last=&n increment=1 border=0 font=12");

rename(name + "-Montage");
```


TAULUKKO 3: OJ_SetScale.ijm

```
// OJ_SetScale.ijm
// Versio 0.5
// Copyright © 2013 Oskari Jääskeläinen

Dialog.create("Set scale");
Dialog.addChoice("Suurennos", newArray("40x", "20x", "10x", "5x",
"2x", "1x"), "20x");
Dialog.show();
s=Dialog.getChoice();

//40x
if (s=="40x") run("Set Scale...", "distance=2000 known=1 pixel=1
unit=mm");
//20x
else if (s=="20x") run("Set Scale...", "distance=1000 known=1 pixel=1
unit=mm");
//10x
else if (s=="10x") run("Set Scale...", "distance=500 known=1 pixel=1
unit=mm");
//5x
else if (s=="5x") run("Set Scale...", "distance=250 known=1 pixel=1
unit=mm");
//2x
else if (s=="2x") run("Set Scale...", "distance=125 known=1 pixel=1
unit=mm");
//1x
else if (s=="1x") run("Set Scale...", "distance=62.5 known=1 pixel=1
unit=mm");
```

TAULUKKO 4: OJ_AREA.IJM

```
// OJ_Area.ijm
// Versio 0.5
// Copyright © 2013 Oskari Jääskeläinen

setTool("freehand");
runMacro("../plugins\\OJ\\OJ_SetScale.ijm");
run("Duplicate...");
run("8-bit");

run("Threshold...");
waitForUser("Set correct threshold and click OK when ready.");

run("Convert to Mask");

waitForUser("Poista artefaktat!");

run("Select None");
run("Set Measurements...", "area limit display redirect=None
decimal=3");

run("Dilate");
run("Dilate");
run("Fill Holes");
run("Dilate");
run("Dilate");
run("Fill Holes");
run("Dilate");
run("Dilate");
run("Fill Holes");
run("Erode");
run("Erode");
run("Erode");
run("Erode");
run("Erode");
run("Erode");

run("Measure");
```

TAULUKOT

Taulukko 1: Immunohistokemialliset vasta-aineet	3
Taulukko 2: OJ_Montage.ijm	37
Taulukko 3: OJ_SetScale.ijm.....	38
Taulukko 4: OJ_Area.ijm.....	39

KUVAT

Kuva 1: JVSView:n lataaminen	10
Kuva 2: JVSView:n lataaminen	10
Kuva 3: JVSView asennus	11
Kuva 4: Hyväksy lisenssi	11
Kuva 5: Oletussijainti.....	12
Kuva 6: JPEG2000 oletusohjelmaksi rekisteröinti	12
Kuva 7: ImageJ asennuspaketin purkaminen	13
Kuva 8: ImageJ asennuspaketin purkaminen	14
Kuva 9: ImageJ asennuspaketin purkaminen	14
Kuva 10: ImageJ asennuspaketin purkaminen	15
Kuva 11: Portable Javan asennus	16
Kuva 12: java -kansion uudelleennimeäminen.....	17
Kuva 13: ImageJ:n automaattinen konfiguraatio	17
Kuva 14: ImageJ:n automaattinen konfiguraatio	18
Kuva 15: ImageJ pääikkuna	18
Kuva 16: Plugins-valikko	19
Kuva 17: JVSView konfigurointi.....	20
Kuva 18: ImageJ:n muistivaraus	20
Kuva 19: ImageJ:n oletusvärit	21
Kuva 20: Binary options.....	21
Kuva 21: "Single instance listener"	21
Kuva 22: JVSView	22
Kuva 23: Vasen yläkulma.....	23
Kuva 24: Vasen reuna.....	24
Kuva 25: Oikea yläkulma	24
Kuva 26: Oikea reuna	24

Kuva 27: Alareuna	24
Kuva 28: Ruudut on kopioitava oikeassa järjestyksessä.....	25
Kuva 29: Käynnistä OJ-Montage.....	26
Kuva 30: Nimeä kuva.....	26
Kuva 31: Syötä sarakkeet ja rivit	26
Kuva 32: Oikein rakennettu montaasi.....	27
Kuva 33: Väärin rakennettu montaasi.....	27
Kuva 34: Thresholding.....	28
Kuva 35: Liian vaalea	29
Kuva 36: Sopiva	29
Kuva 37: Liian tumma.....	29
Kuva 38: Ympäroi artefakta.....	30
Kuva 39: Paina Ctrl+X	31
Kuva 40: Näytteen pinta-alat	32